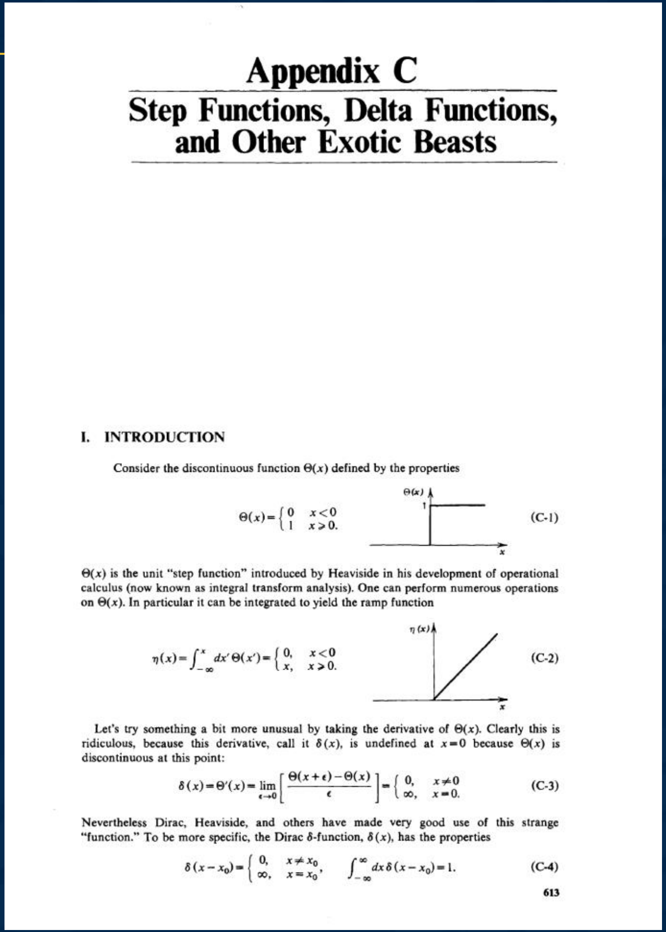
MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Software Engineering Practices in Scientific Computing

and other exotic beasts

Prof. Kochunas

EECS 481 (W23)



Appendix C
Step Functions, Delta Functions, and Other Exotic Beasts

I. INTRODUCTION

Consider the discontinuous function $\Theta(x)$ defined by the properties

$$\Theta(x)=\begin{cases}0 & x<0\\ 1 & x\geq 0.\end{cases} \qquad \text{(C-1)}$$

$\Theta(x)$ is the unit "step function" introduced by Heaviside in his development of operational calculus (now known as integral transform analysis). One can perform numerous operations on $\Theta(x)$. In particular it can be integrated to yield the ramp function

$$\eta(x)=\int_{-\infty}^{x}dx'\,\Theta(x')=\begin{cases}0, & x<0\\ x, & x\geq 0.\end{cases} \qquad \text{(C-2)}$$

Let's try something a bit more unusual by taking the derivative of $\Theta(x)$. Clearly this is ridiculous, because this derivative, call it $\delta(x)$, is undefined at $x=0$ because $\Theta(x)$ is discontinuous at this point:

$$\delta(x)=\Theta'(x)=\lim_{\epsilon\to 0}\left[\frac{\Theta(x+\epsilon)-\Theta(x)}{\epsilon}\right]=\begin{cases}0, & x\neq 0\\ \infty, & x=0.\end{cases} \qquad \text{(C-3)}$$

Nevertheless Dirac, Heaviside, and others have made very good use of this strange "function." To be more specific, the Dirac δ-function, $\delta(x)$, has the properties

$$\delta(x-x_0)=\begin{cases}0, & x\neq x_0\\ \infty, & x=x_0\end{cases}, \qquad \int_{-\infty}^{\infty}dx\,\delta(x-x_0)=1. \qquad \text{(C-4)}$$

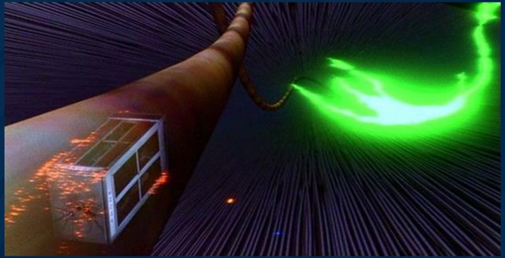
613

One-Slide Summary

- Prof. Kochunas 5-steps to computational science software development
  1. Make it work (get it to run)
  2. Make it right (satisfy functional requirements)
  3. Make it robust (get rid of bugs)
  4. Make it fast (get answer quicker)
  5. Make it usable (should not need PhD to use)

Outline

- Multi-Language Programs
- Requirements
- Design for Maintainability
- Testing Quality Metrics
- QA Processes
- Test Inputs and Oracles
- Fun with VR (sort of)



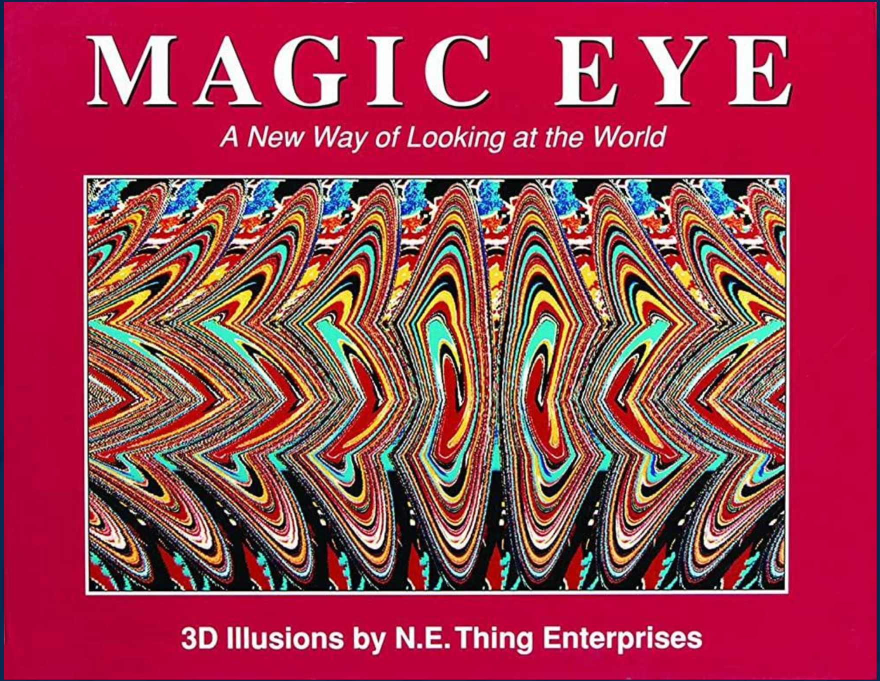
MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Learning Objectives: by the end of today's lecture you should be able to…

1. (*value*) a lot of the course topics have immense practicality

2. (*value*) software development in computational science is probably more fun than working at a tech company

Background


MAGIC EYE
A New Way of Looking at the World
3D Illusions by N.E. Thing Enterprises

Wait… why is a nuclear engineer here?

- Software from my PhD has
  - R&D 100 (2016)
  - NQA-1 Certified (2019)
  - Supported ~14 PhDs
- Also Developed graduate course "Methods and Practice of Scientific Computing" for MICDE in 2016

CASL
A DOE Energy Innovation Hub

Consortium for the Advanced Simulation of LWRs (10 years and $250M)

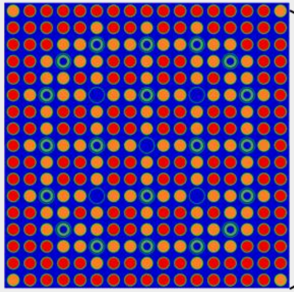
More specifically…

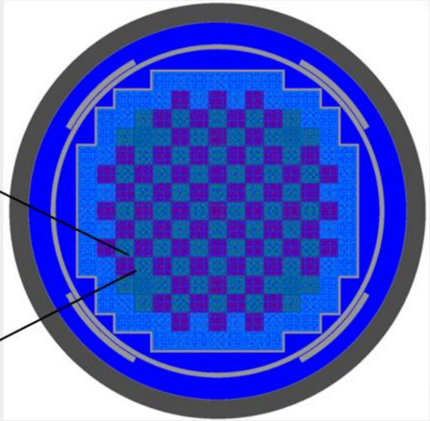
Watts Bar
Unit 1

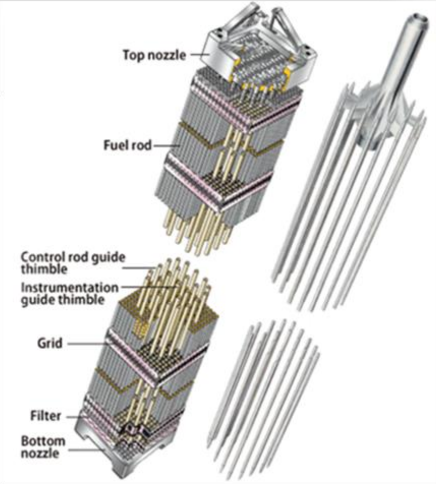
Solve This:

$$\vec{\Omega}\cdot\nabla\varphi\left(\vec{r},\vec{\Omega},E\right)+\Sigma_t\left(\vec{r},E\right)\varphi\left(\vec{r},\vec{\Omega},E\right)=$$
$$\frac{\chi(E)}{4\pi k_{eff}}\int_0^{\infty}\nu\Sigma_f\left(\vec{r},E'\right)\int_0^{4\pi}\varphi\left(\vec{r},\vec{\Omega}',E'\right)d\Omega' dE'+$$
$$\int_0^{\infty}\int_0^{4\pi}\Sigma_s\left(\vec{r},\vec{\Omega}'\cdot\vec{\Omega},E'\rightarrow E\right)\varphi\left(\vec{r},\vec{\Omega}',E'\right)d\Omega' dE'$$

For that:

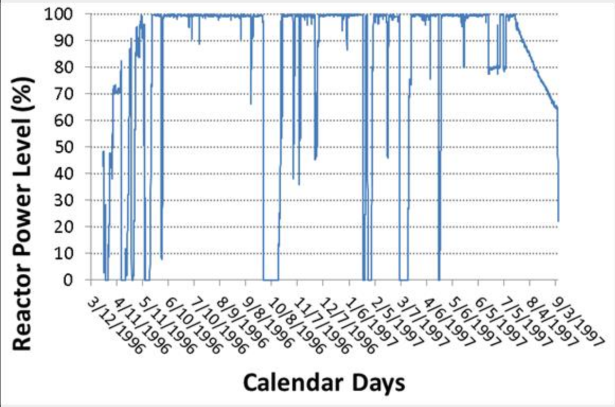


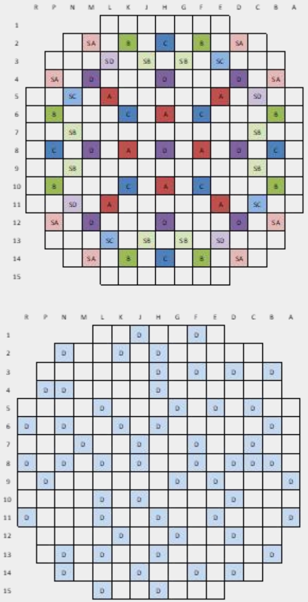




Oh by the way…

It operates like this:





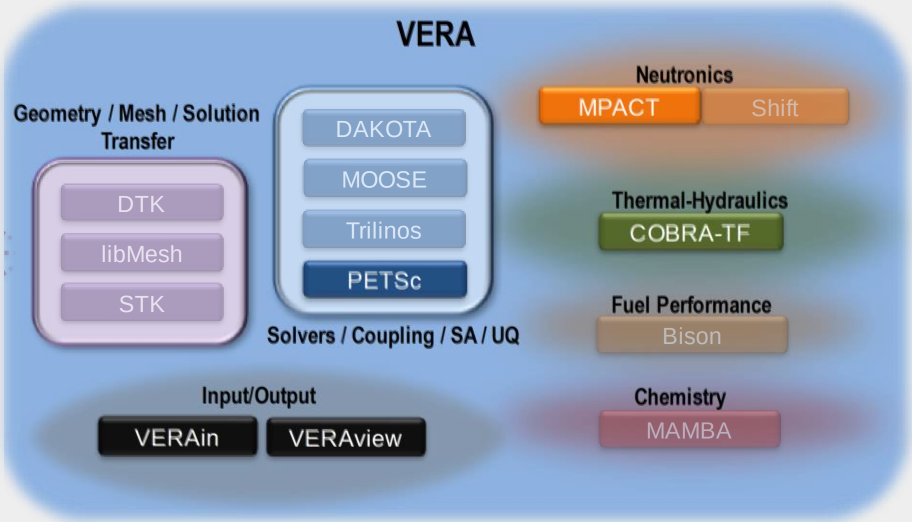
And you'll need this:

- Largest open science computer in U.S.

but really for industry they want to use 1000 cores and deplete a cycle overnight

Michigan PArallel Characteristics based Transport

- MPACT *did not exist at the outset* of CASL, but *grew out of the program*.
- **To CASL**: MPACT is *the* deterministic neutronics code to solve the pin resolved power distribution throughout the reactor core.
  - Sits at the heart of the "core simulator" capability.
- **To UM**: MPACT is a research tool designed in a flexible way to facilitate PhD research in several areas
  - Transport methods, acceleration methods, parallel algorithms
  - Reactor Analysis and multi-physics numerical methods
  - As a teaching tool



The VERA Core Simulator

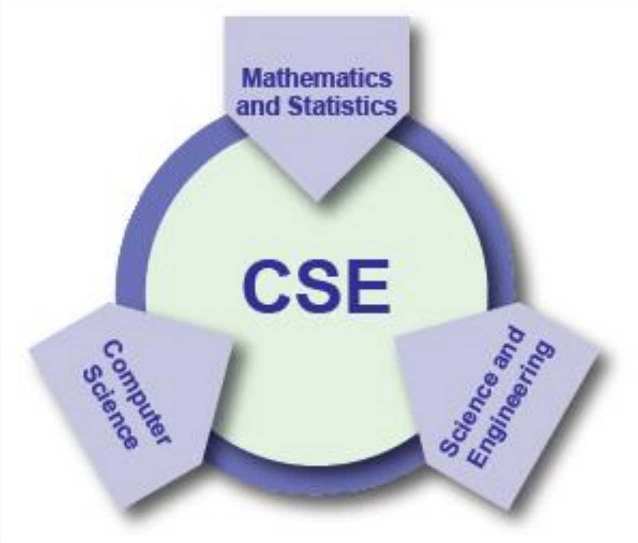
Spoiler alert! We did it!

at the heart of this is

MPACT®

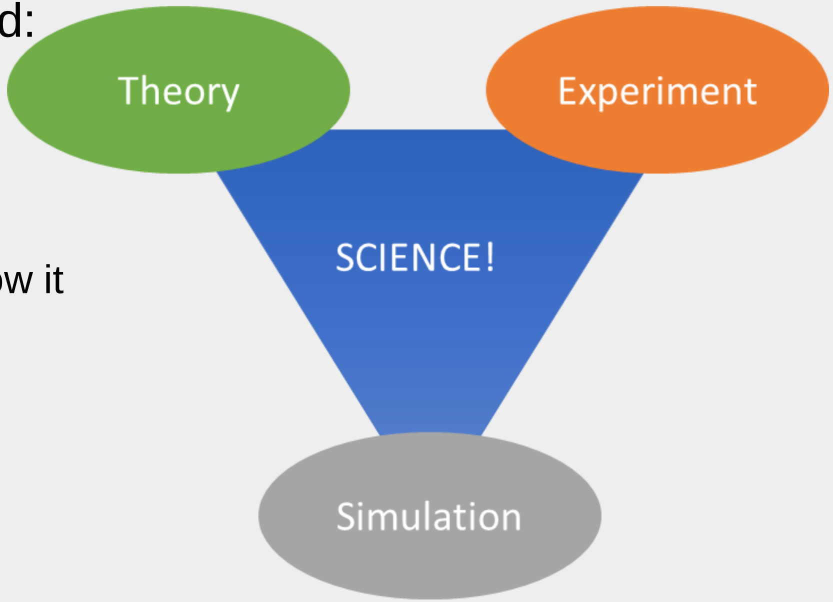
What is Computational Science and Engineering?

- CSE is a recently established multidisciplinary field of research and education
  - It lies at the intersection of mathematics, computer science, and science & engineering.
- As engineers how do we participate in CSE?
  - We have the applications & technology
  - We understand the physics governing our systems
  - But we may not know the latest math or how to effectively utilize computers to solve our problems



Pillars of Science

- Why so much focus on CSE?
- Traditional scientific and engineering method:
  - 1. Do theory or paper design
  - 2. Perform experiments, build prototypes, etc.
- Limitations
  - Too difficult—build a large wind tunnel
  - Too expensive—build a passenger jet and throw it away
  - Too dangerous—nuclear weapons
  - Too slow—climate change or astral evolution
- Computational science and engineering paradigm
  - 3. Use computers to <u>*simulate and analyze*</u> phenomenon



PhD’s at U of M using MPACT

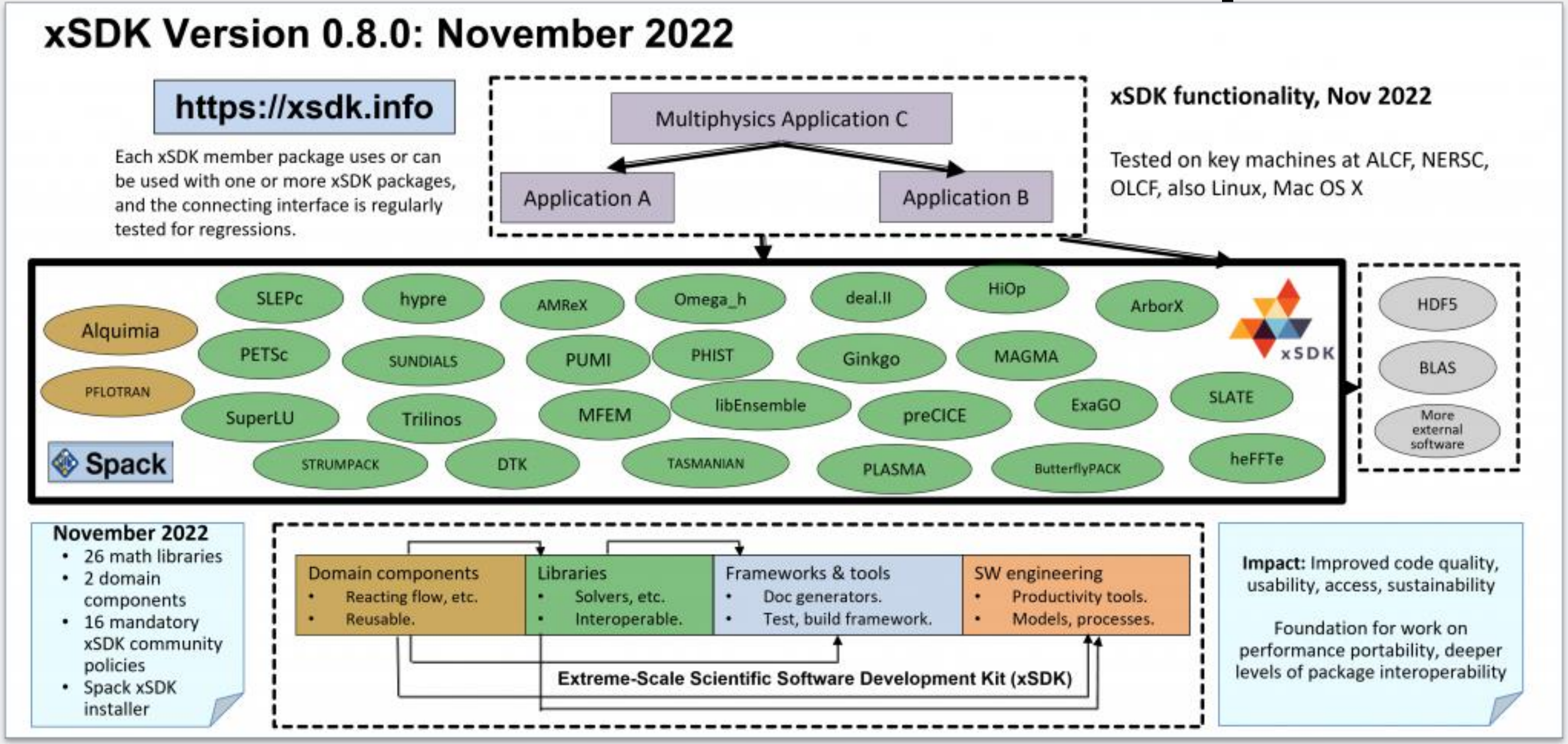
- **Brendan Kochunas, 2013** – A Hybrid Parallel Algorithm for the 3-D Method of Characteristics Solution of the Boltzmann Transport Equation on High Performance Compute Clusters http://hdl.handle.net/2027.42/100072
- **Travis Trahan, 2014** – An Asymptotic, Homogenized, Anisotropic, Multigroup Diffusion Approximation to the Neutron Transport Equation http://hdl.handle.net/2027.42/107152
- **Blake Kelley, 2015** – An Investigation of 2D/1D Approximations to the 3D Boltzmann Transport Equation http://hdl.handle.net/2027.42/113576
- **Yuxuan Liu, 2015** – A Full Core Resonance Self-shielding Method Accounting for Temperature-dependent Fuel Subregions and Resonance Interference http://hdl.handle.net/2027.42/111419
- **Shane Stimpson, 2015** – An Azimuthal Fourier Moment-Based Axial SN Solver for the 2D/1D Scheme http://hdl.handle.net/2027.42/111446
- **Thomas Saller, 2015** – Asymptotic Homogenized SP2 Approximations to the Neutron Transport Equation http://hdl.handle.net/2027.42/116754
- **Ang Zhu, 2016** – Transient Methods for Pin-Resolved Whole Core Transport http://hdl.handle.net/2027.42/133353
- **Dan Walter, 2016** – A High Fidelity Multiphysics Framework for Modeling CRUD Deposition on PWR Fuel Rods http://hdl.handle.net/2027.42/120638
- **Mitchell Young, 2016** – Orthogonal-Mesh, 3D Sn with Embedded 2-D Method of Characteristics for Whole-Core, Pin-Resolved Reactor Analysis http://hdl.handle.net/2027.42/135759
- **Michael Rose, 2017** – Multiphysics Simulation of Fission Gas Production and Release in Light Water Reactor Fuel http://hdl.handle.net/2027.42/140807
- **Aaron Graham, 2017** – Subgrid Methods for Resolving Axial Heterogeneity in Planar Synthesis Solutions for the Boltzmann Transport Equation http://hdl.handle.net/2027.42/138586
- **Benjamin Yee, 2018** – A Multilevel in Space and Energy Solver for Multigroup Diffusion and Coarse Mesh Finite Difference Eigenvalue Problems http://hdl.handle.net/2027.42/146075
- **Michael Jarrett, 2018**– A 2D/1D Neutron Transport Method with Improved Angular Coupling http://hdl.handle.net/2027.42/147498

Scientific Computing is inherently multi-lingual

Multi-language Projects: One-Slide Summary

- Many modern software projects involve code written in **multiple languages**. This can involve a common **bytecode** or **C native** method interfaces.
- Native code interfaces can be understood in terms of (1) **data layout** and (2) special common **functions to manipulate** managed data.
- Almost all aspects of software engineering are impacted in multi-language projects.

xSDK – eXascale Software Development Kit



Languages PETSc

- C 78.3%
- C++ 6.1%
- Python 5.5%
- Cython 3.1%
- Perl 2.0%
- Fortran 1.8%
- Other 3.2%

Languages SUNDIALS

- C 65.5%
- Fortran 14.1%
- C++ 9.5%
- CMake 4.6%
- Shell 1.9%
- Cuda 1.8%
- Other 2.6%

Languages heFFTe

- C++ 66.5%
- Fortran 24.7%
- CMake 4.5%
- Python 1.6%
- Cuda 1.2%
- C 1.2%
- Other 0.3%

Languages SuperLU

- C 96.1%
- MATLAB 1.0%
- CSS 0.9%
- CMake 0.9%
- Makefile 0.7%
- Fortran 0.3%
- Other 0.1%

Languages PFLOTRAN

- Fortran 85.8%
- C++ 5.1%
- Python 4.3%
- Gnuplot 1.4%
- TeX 1.1%
- Makefile 1.0%
- Other 1.3%

Languages MAGMA

- C++ 54.0%
- Fortran 27.4%
- Cuda 13.8%
- C 4.2%
- Python 0.3%
- Makefile 0.2%
- Other 0.1%

Languages STRUMPACK

- C++ 81.1%
- Fortran 9.1%
- CMake 4.8%
- Cuda 1.6%
- C 1.3%
- Shell 1.1%
- Other 1.0%

Languages OpenBlas

- C 51.2%
- Fortran 30.5%
- Assembly 16.8%
- Makefile 0.5%
- C++ 0.4%
- CMake 0.4%
- Other 0.2%

Languages Spack

- Python 98.4%
- Shell 0.9%
- Prolog 0.3%
- C 0.2%
- CMake 0.1%
- Makefile 0.1%

Computational Science Requirements are more than functional

Requirements: One-Slide Summary

- **Requirements** articulate the relationship and interface between a desired system and its environment. This includes both what is (or is expected) and what should be.
- We distinguish between **functional** and **quality** (or non-functional) requirements. Both should be stated in measurable ways.
- Requirements can describe variables, inputs, and outputs, and assumptions between them.
- We distinguish between informal statements and **verifiable** requirements.

High-Level Requirements (Roadmap)

Date Completed	Benchmark Problem Description
09/2012	#1 2D HZP Pin Cell
12/2012	#2 2D HZP Lattice
~5/2013	#3 3D HZP Assembly
~7/2013	#4 HZP 3x3 Assembly CRD Worth
11/2013	**#5 Physical Reactor Zero Power Physics Tests (ZPPT)**
11/2013	#6 HFP BOL Assembly
4/2014	#7 HFP BOC Physical Reactor w/ Xenon
7/2014	**#8 Physical Reactor Startup Flux Maps**
9/2014	**#9 Physical Reactor Depletion**
2/2015	#10 Physical Reactor Refueling

Application drives development

A cycle depletion on < 1000 cores, overnight

MPACT declared CASL deterministic pin resolved neutronics tool. ORNL becomes co-developer

Functional Requirements

- What do we notice?
- What's good/bad about these requirements

Fuel Shuffle File Requirements

1. Must implement the "VERAIn Specification for LWR Core Shuffling and Jump-in" specifications below for Cartesian grid LWRs. Specifically:
   a. Must be able to process the specified VERA input options as stated
   b. Must be able to read in one or more (depending on mode) properly generated VERA restart files and use them to begin a calculation
   c. Must be able to produce the specified restart file at the end of a VERA state calculation
2. Must be able to write, read, and shuffle (where applicable) any reactor component of any reactor type for which tracking irradiation may be of interest:
   a. Fuel rods/assemblies
   b. Inserts
   c. Control rods/blades
   d. Nozzles
   e. Core plates
   f. Grids
   g. Baffle/reflector
   h. Reactor vessel
   i. Composite blocks
   j. Fuel pebbles
   k. Molten salt compositions
   l. Solid moderator components (e.g. graphite blocks)
3. Must work for all reactor types
4. Must be able to write, read, and shuffle (when appropriate) the following information:
   a. Geometry information: Core layout, axial mesh, thermal expansion parameters
   b. Feedback and post operation data
      i. Component-specific data (e.g. vessel fluence from SHIFT)
      ii. Pin-wise data (CRUD/corrosion data)
      iii. Any other feedback/post operation data that could be required
   c. State data such as power history and reactor state data from the end of the previous calculation (e.g. control rod positions, boron concentration, etc.)
5. Must support reconstituted rods for solid-fueled reactors when performing restarts or shuffles
6. Must be able to restart using different thermal expansion temperatures than were used to

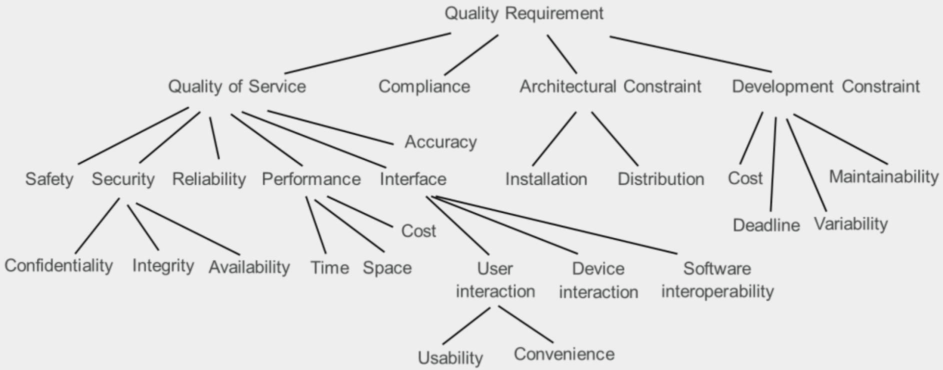
Need to talk to [redacted] about the exact requirements for this (I think I have a good guess but I'll make sure with him), then the developers can determine what information to write to the file to accomplish that

Quality Requirements

Input Example: Write and Read a Restart File

```
[STATE]
  deplete EFPD 0.0
  restart_write  restart_cyc12.h5 “BOC”
[STATE]
  deplete EFPD 20 40 80 100
  restart_write  restart_cyc12.h5 “100EFPD”
[STATE]
  deplete EFPD 150 200
  restart_write  restart_cyc12.h5 “200EFPD”
```

```
[STATE]
  power 50.0
  boron 800
  restart_read restart_cyc12.h5 “200EFPD”
```



7. Should be able to optionally write & read depletion isotopes in addition to transport isotopes
8. Must support various levels of HDF5 compression
9. Must support axial remeshing on an assembly-by-assembly basis
10. Must support radial remeshing on a pin-by-pin basis

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

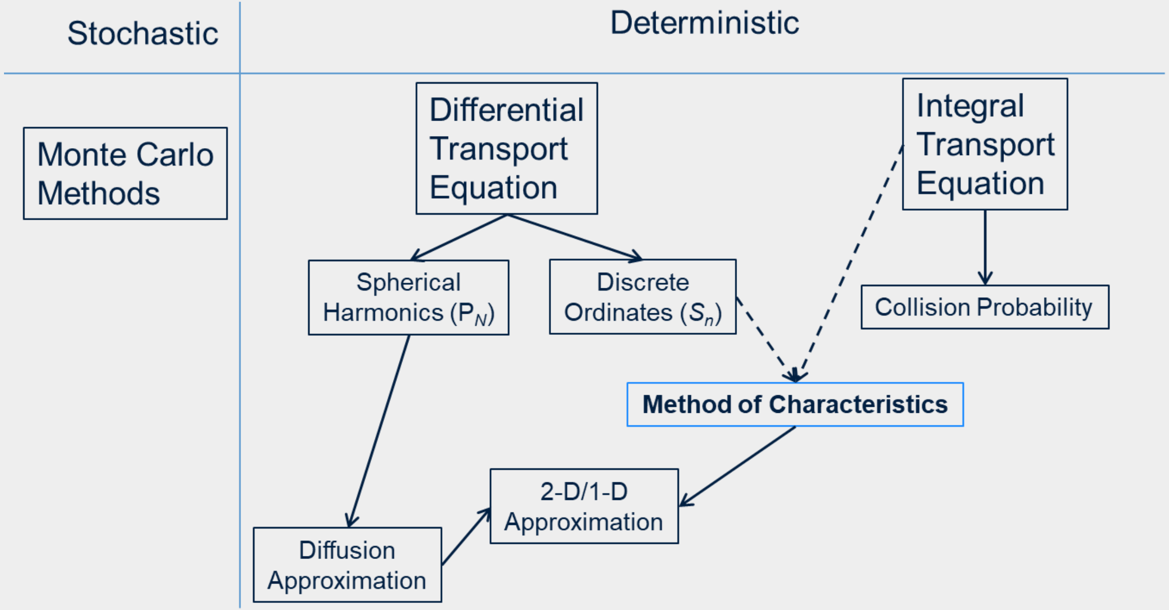
Design for Maintainability

Design: One-Slide Summary

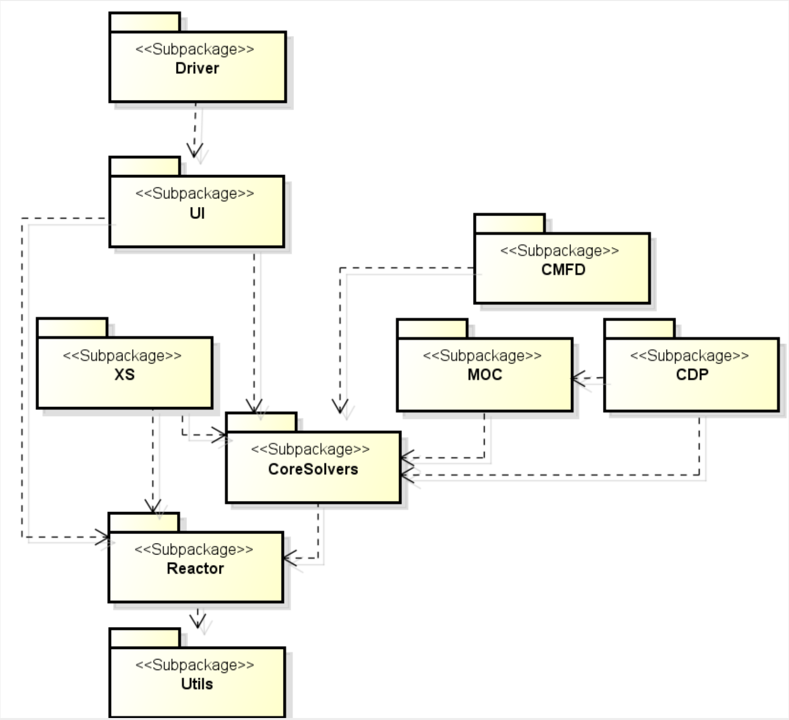
- We can invest up-front effort in **designing** software to facilitate **maintenance** activities. This reduces overall lifecycle costs.
- We will consider designing to improve comprehension, documentation, change, reuse, and testability.
  - The *metrics* used for understandability, the category of information conveyed by documentation, object-oriented principles and design patterns, and coverage are all relevant.

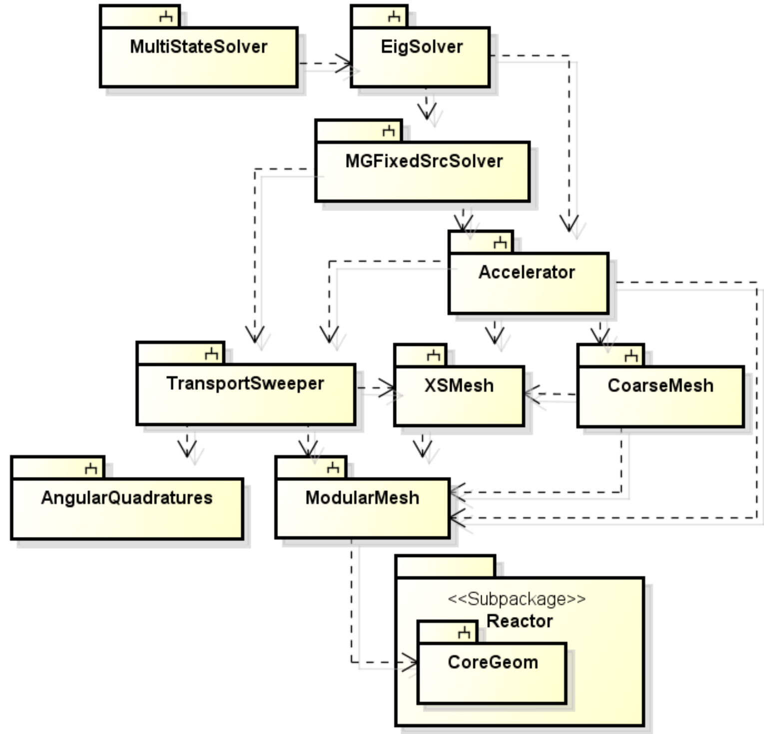
Approaches to Solving the Boltzmann Neutron Transport Equation

$$\vec{\Omega}\cdot\nabla\varphi(\vec{r},\vec{\Omega},E)+\Sigma_t(\vec{r},E)\varphi(\vec{r},\vec{\Omega},E)=\frac{\chi(E)}{4\pi k_{eff}}\int_0^\infty \nu\Sigma_f(\vec{r},E')\int_0^{4\pi}\varphi(\vec{r},\vec{\Omega}',E')d\Omega' dE'+\int_0^\infty\int_0^{4\pi}\Sigma_s(\vec{r},\vec{\Omega}'\cdot\vec{\Omega},E'\to E)\varphi(\vec{r},\vec{\Omega}',E')d\Omega' dE'$$

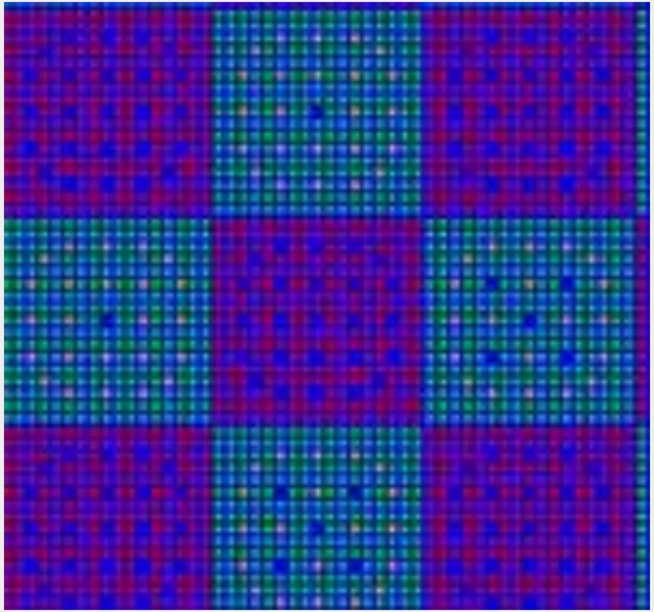


Dependencies and Class Hierarchy

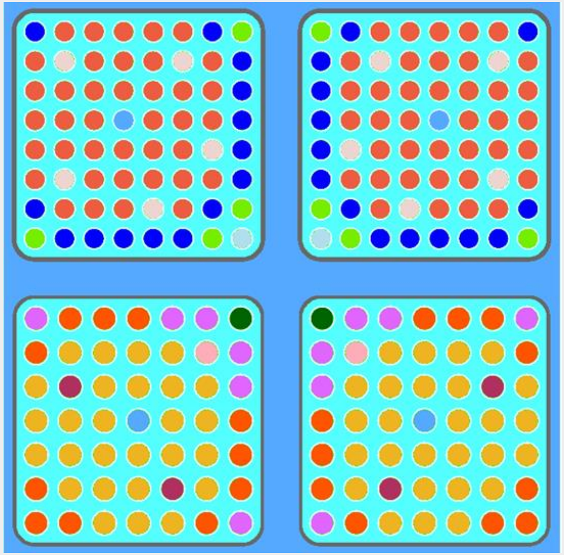




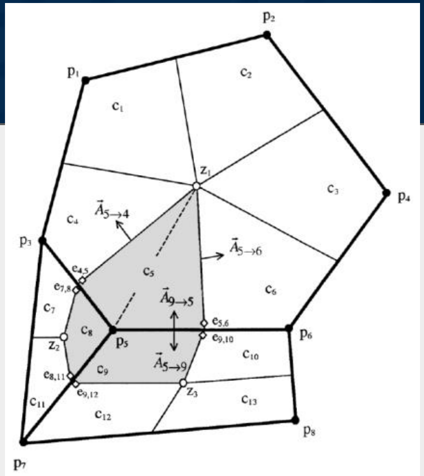
Geometry and PWR to BWR

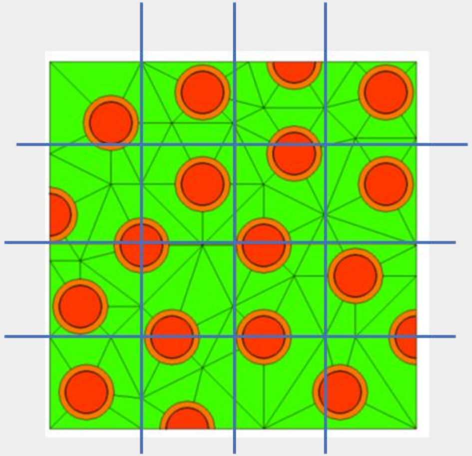


Pressurized Water Reactor (PWR) Assemblies

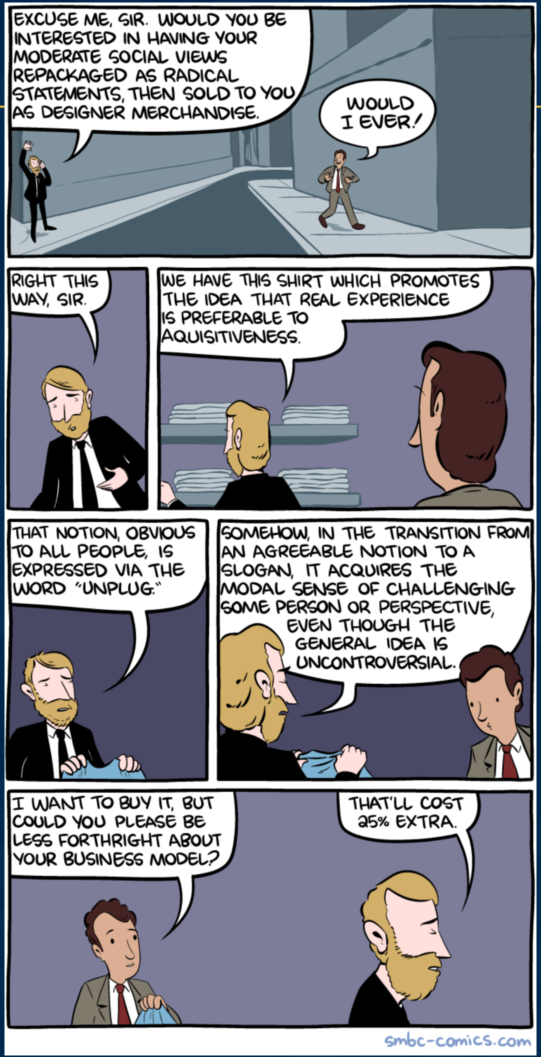


Boiling Water Reactor (BWR) Assemblies (8x8 and 7x7)





Testing Quality Metrics



QA and Testing: One-Slide Summary

- **Quality Assurance** maintains desired product properties through process choices.
- **Testing** involves running the program and inspecting its results or behavior. It is the dominant approach to software quality assurance. There are numerous methods of testing, such as regression testing, unit testing, and integration testing.
- **Mocking** uses simple replacement functionality to test difficult, expensive, or unavailable modules or features.

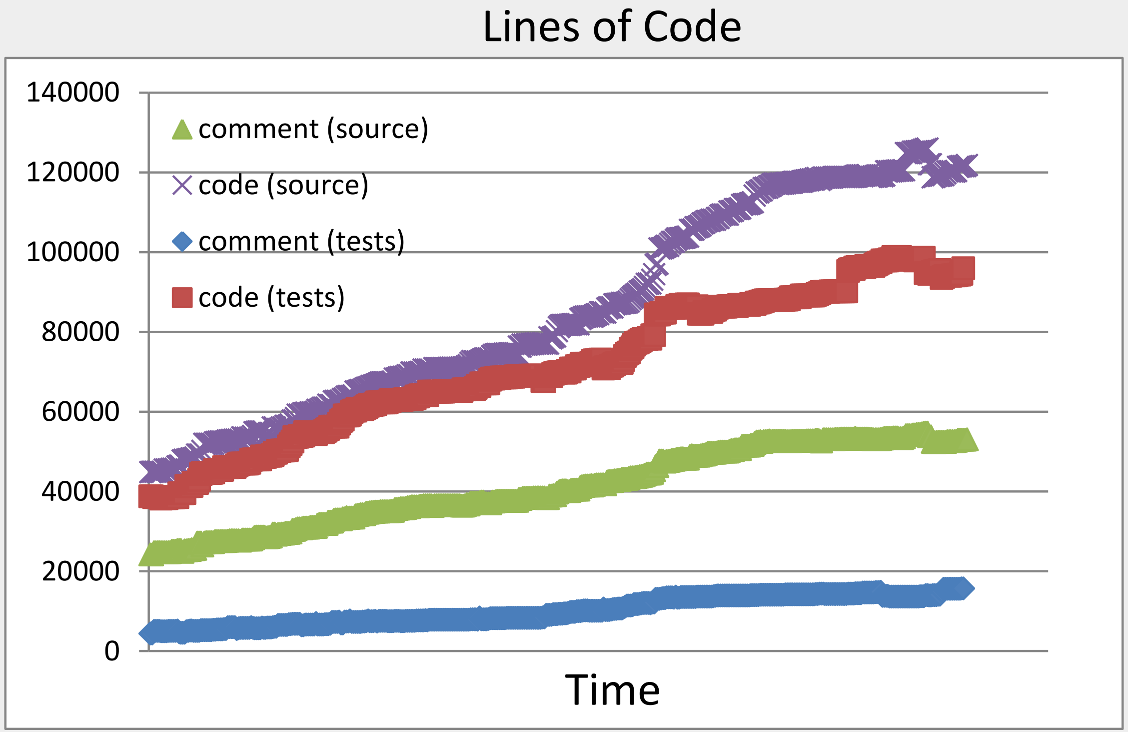
(special thanks to James Perretta for material)

Test Quality Metrics: One-Slide Summary

- **Test suite quality metrics** help us decide which suite to use. **Line coverage**, the fraction of lines visited when running a suite, is simple but gives limited confidence.
- **Branch coverage**, which requires both true and false values for conditions, is richer (incorporating data values indirectly).
- **Mutation analysis** measures the fraction of seeded defects detected by a suite; it is expensive but effective.
- **Beta** and **A/B testing** involve real users and their experiences.

Test and Source Code Metrics

- Code Coverage (by line)
  - Utils: 91.6%
  - Reactor: 84.0%
  - CoreSolvers: 84.1%
  - MOC: 70.3%
  - CMFD: 28.9%
  - XS: 73.5%
  - UI: 57.9%



Integration and Regression Testing

Geometry	
	Quarter
	Full
	Assembly Gap
	Nozzles
	Core Plates
	Reflector
	Baffle
	Inserts
	Control Rods
	Detector
	Nominal Spacers
	Above/below fuel
	IFM
	Explicit

Solvers	
	2D P0
	2D Pn
	2D/1D NEM
	2D/1D SP3
	CMFD 1g
	CMFD MG
	Search keff
	Search boron
	Search rod
	Multistate
	Depletion native
	Depletion origen
	Shuffle rotational
	Shuffle mirror
	InternalTH
	CTF
	Eq Xe/Sm
	Subgroup
	ESSM
	Cusping

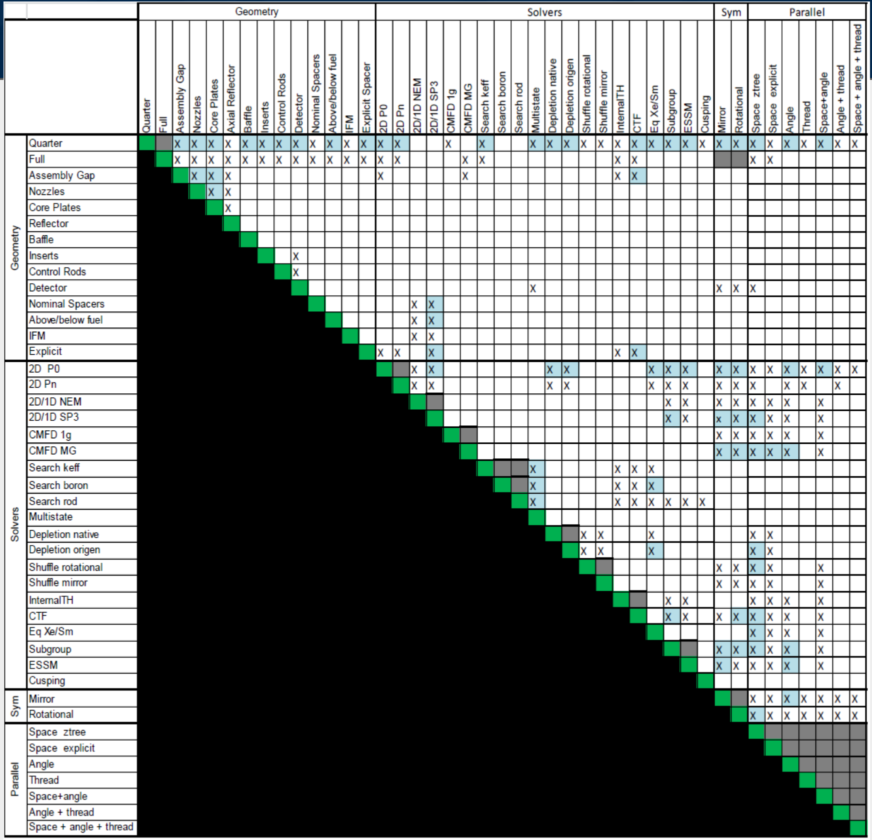
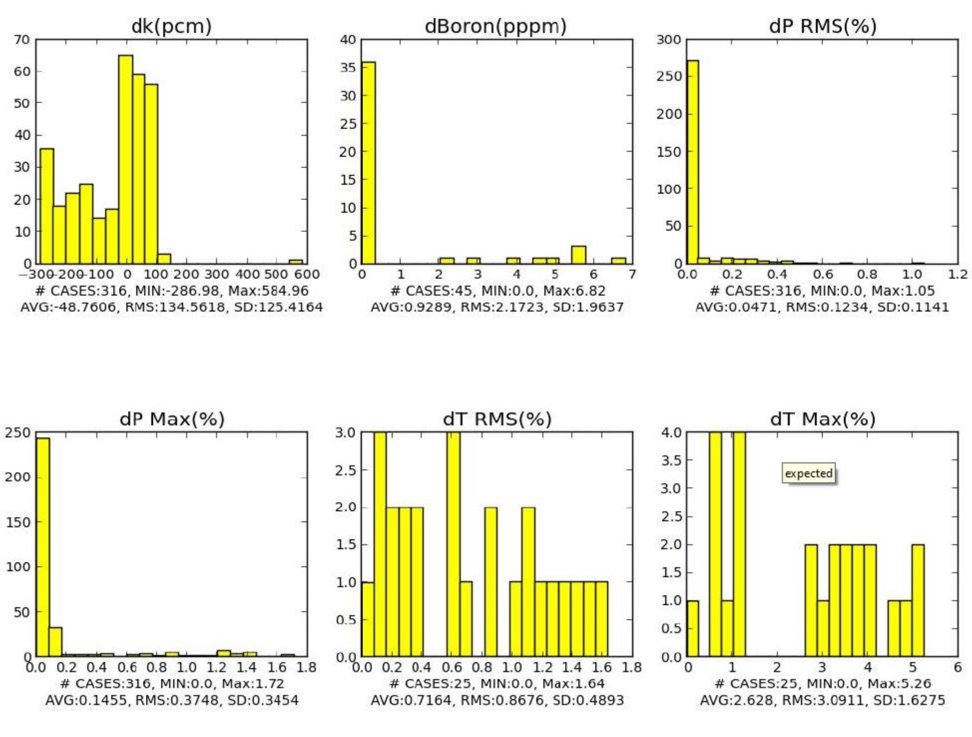
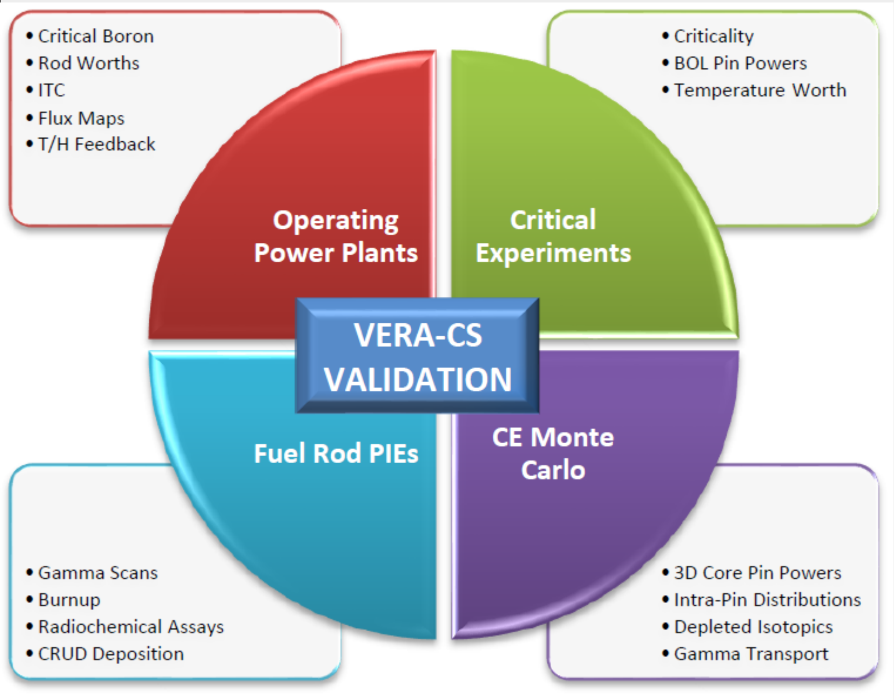


Figure 2.4 MPACT Regression Suite Matrix [Collins, 2015a].

Regression Test Suite Acceptance



Validation Testing (Test against reality)



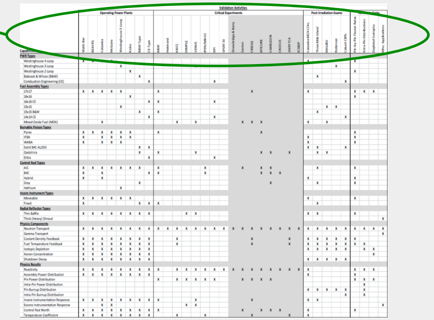
Validation Activities

	Operating Power Plants								Critical Experiments																Post-Irradiation Exams					CE Monte Carlo			
Capabilities	Watts Bar	BEAVRS	Catawba	McGuire	Westinghouse 3-Loop	Krsko	B&W-Type	CE-Type	B&W	Helstrand	KRITZ	DIMPLE	VENUS	IPEN/MB-01	RPI	SPERT III	Strawbridge & Barry	Saxton	CREOLE	EPICURE	CAMELEON	CROCUS	JAERI TCA	ICSBEP	Catawba MOX LTAs	Three Mile Island	MALIBU	Robinson	Calvert Cliffs	Pin-by-Pin Fission Rates	Intra-Pin Distributions	Depleted Isotopics	Misc. Applications
PWR Types																																	
Westinghouse 4-Loop	X	X	X	X																					X					X			
Westinghouse 3-Loop					X								X															X					
Westinghouse 2-Loop						X																								X			
Babcock & Wilcox (B&W)							X																			X							
Combustion Engineering (CE)								X							X														X				
Fuel Assembly Types																																	
17x17	X	X	X	X	X				X										X						X					X	X	X	
16x16						X																								X			
16x16 CE								X	X						X																		
15x15					X																						X	X					
15x15 B&W							X		X																	X							
14x14 CE								X							X														X				
Mixed Oxide Fuel (MOX)			X								X		X					X	X	X					X		X						
Burnable Poison Types																																	
Pyrex	X	X	X	X	X	X			X											X										X			
IFBA	X		X	X	X	X																			X					X			
WABA	X		X	X	X																				X					X			
Solid B4C-AL2O3							X	X																	X	X			X				
Gadolinia							X		X				X							X	X		X			X	X						
Erbia								X							X																		
Control Rod Types																																	
AIC	X	X	X	X	X	X	X		X					X				X		X	X					X				X			
B4C	X		X					X	X					X						X	X	X											
Hybrid	X		X																						X					X			
Gray							X														X					X				X			
Hafnium					X																												
Incore Instrument Types																																	
Moveable	X	X	X	X	X	X																			X					X			
Fixed	X						X	X	X																	X							
Radial Reflector Types																																	
Thin Baffle	X	X	X	X	X	X	X	X				X	X												X	X				X			
Thick (Heavy) Shroud																																	X
Physics Components																																	
Neutron Transport	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Gamma Transport																																	X
Coolant Density Feedback	X	X	X	X	X	X	X	X			X								X				X		X	X	X	X	X	X			
Fuel Temperature Feedback	X	X	X	X	X	X	X	X			X								X				X		X	X	X	X	X	X	X		
Isotopic Depletion	X	X	X	X	X	X	X	X																	X	X	X	X	X		X	X	
Xenon Concentration	X	X	X	X	X	X	X	X																								X	
Shutdown Decay	X	X	X	X	X	X	X	X																	X	X	X	X	X				
Physics Results																																	
Reactivity	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	
Assembly Power Distribution	X	X	X	X	X	X	X	X																	X					X			
Pin Power Distribution									X		X	X	X	X	X			X	X	X	X									X	X	X	
Intra-Pin Power Distribution																															X		
Pin Burnup Distribution																									X	X	X	X	X		X		
Intra-Pin Burnup Distribution																													X		X		
Incore Instrumentation Response	X	X	X	X	X	X	X	X	X				X						X						X	X							
Excore Instrumentation Response			X	X								X	X																				X
Control Rod Worth	X	X	X	X	X	X	X	X	X									X		X	X	X			X					X			
Temperature Coefficient	X	X	X	X	X	X	X	X			X			X					X				X		X								

Map Functional Requirements Validation Data

Validation Activities

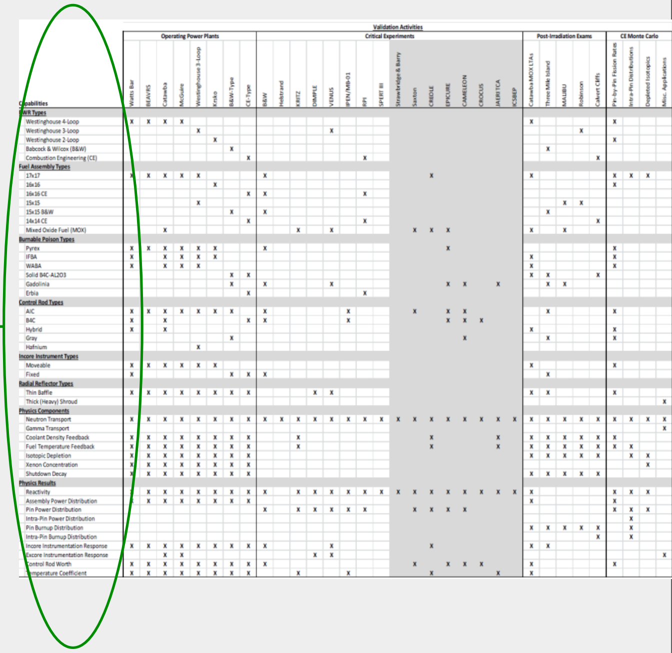
Operating Power Plants								Critical Experiments																Post-Irradiation Exams					CE Monte Carlo			
Watts Bar	BEAVRS	Catawba	McGuire	Westinghouse 3-Loop	Krsko	B&W-Type	CE-Type	B&W	Helstrand	KRITZ	DIMPLE	VENUS	IPEN/MB-01	RPI	SPERT III	Strawbridge & Barry	Saxton	CREOLE	EPICURE	CAMELEON	CROCUS	JAERI TCA	ICSBEP	Catawba MOX LTAs	Three Mile Island	MALIBU	Robinson	Calvert Cliffs	Pin-by-Pin Fission Rates	Intra-Pin Distributions	Depleted Isotopics	Misc. Applications



Map Functional Requirements Validation Data

Capabilities
PWR Types
Westinghouse 4-Loop
Westinghouse 3-Loop
Westinghouse 2-Loop
Babcock & Wilcox (B&W)
Combustion Engineering (CE)
Fuel Assembly Types
17x17
16x16
16x16 CE
15x15
15x15 B&W
14x14 CE
Mixed Oxide Fuel (MOX)
Burnable Poison Types
Pyrex
IFBA
WABA
Solid B4C-AL2O3
Gadolinia
Erbia

Physics Components
Neutron Transport
Gamma Transport
Coolant Density Feedback
Fuel Temperature Feedback
Isotopic Depletion
Xenon Concentration
Shutdown Decay



We still have bugs and get the wrong answer

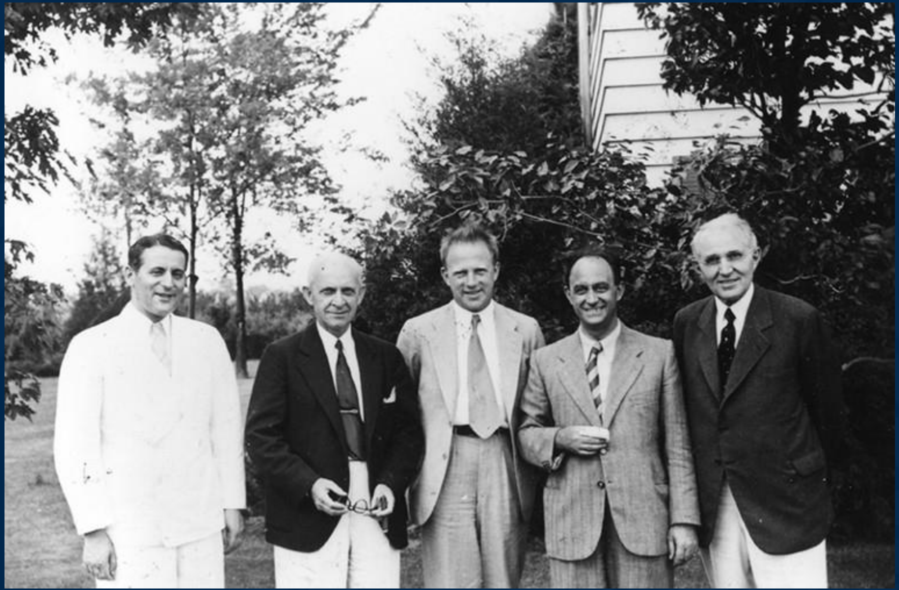
Trivia Break

Trivia: Physicist

- This Dutch-American physicist is credited with jointly proposing the concept of electron spin at 23. He was the editor-in-chief of the leading physics journal Physical Review Letters, received the National Medal of Science, and has a named collection of Egyptian antiquities

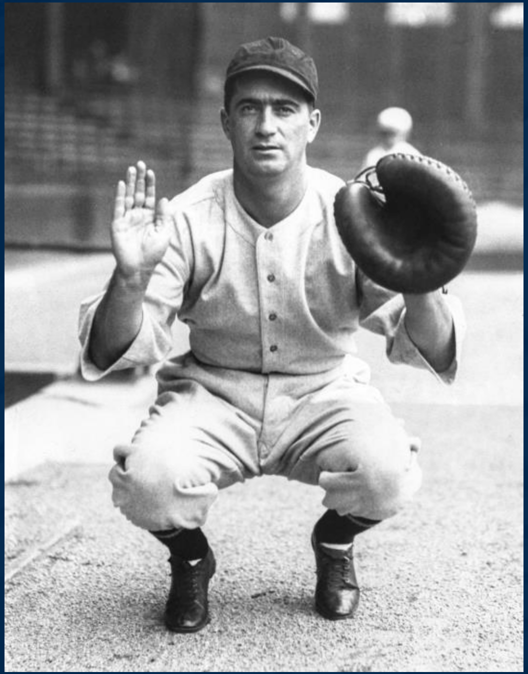
Trivia: Great Minds

- These photo features
  - Samuel Goudsmit
  - Werner Heisenberg
  - Enrico Fermi
- It was taken at one of the world’s pre-eminent physics summer schools in 1939
- Where was it taken?



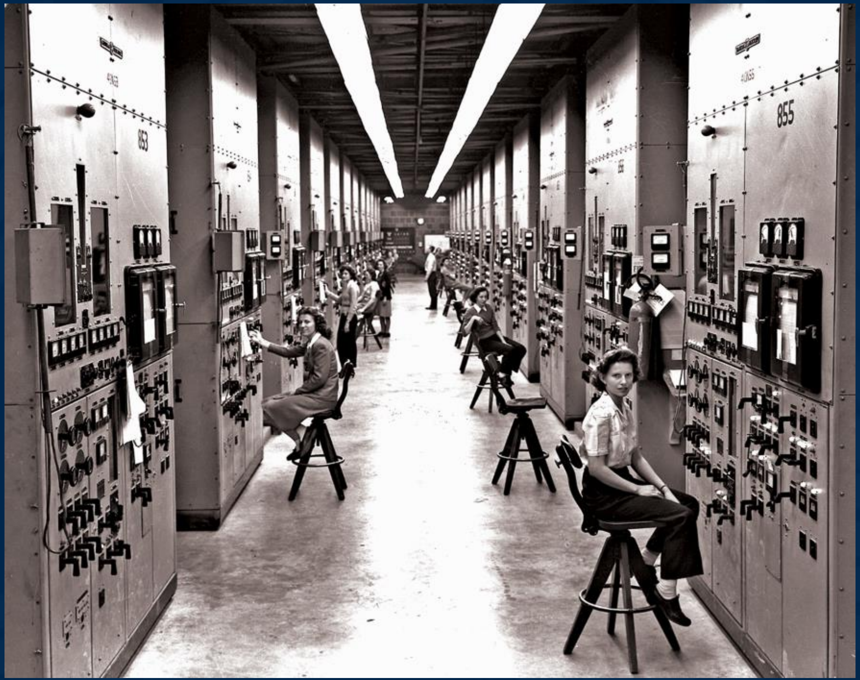
Trivia: The Catcher is the Spy

- This Jewish American who's parents emigrated from Ukrain was professional baseball player spoke seven languages and regularly read 10 newspapers a day (in various languages). He also conducted paramilitary operations for the Office of Strategic Services (predecessor to the CIA). During WWII he provided crucial intelligence on Japan providing video footage of Tokyo.

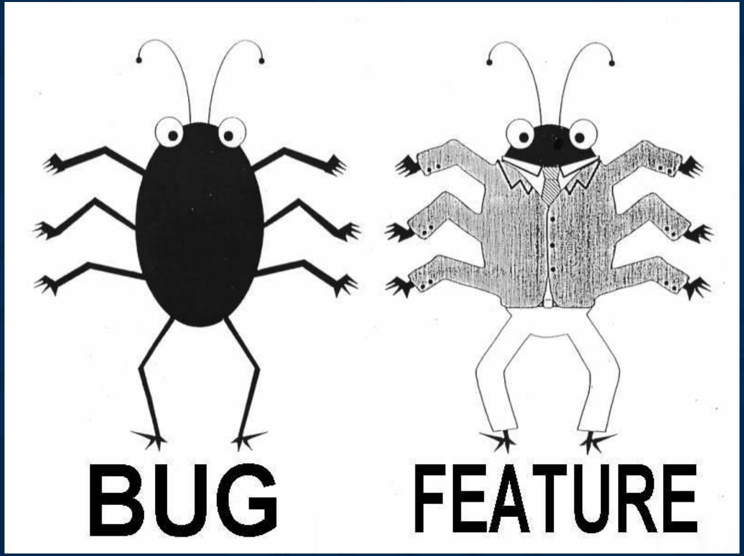


Trivia: Manhattan Project

- These women operated this type of machine who's inventor received a Nobel prize (in 1939) for its invention.
- Despite the women not being told exactly what it was they were operating. They engaged in a week long competition with male scientists (mostly PhDs) to see who could produce more product and outperformed their male colleagues.
- Their service was crucial to the production of "tube alloy" for "Little Boy"



QA Processes



Inputs & Oracles: One-Slide Summary

- Formally, a **test case** consists of an **input** (**data**), an **oracle** (output), and a **comparator**.
- Test inputs determine the behavior of the program. High-coverage inputs can be **generated automatically** through path enumeration, path predicates, and mathematical constraint solving.
- Test oracles correspond to what the program should do. Generating them is an expensive problem; but it can be done automatically (sort of) through invariants and mutation.
- **Test suite minimization** finds the smallest subset of tests that meet a coverage goal.

Ideal Maturity of Software Quality Metrics

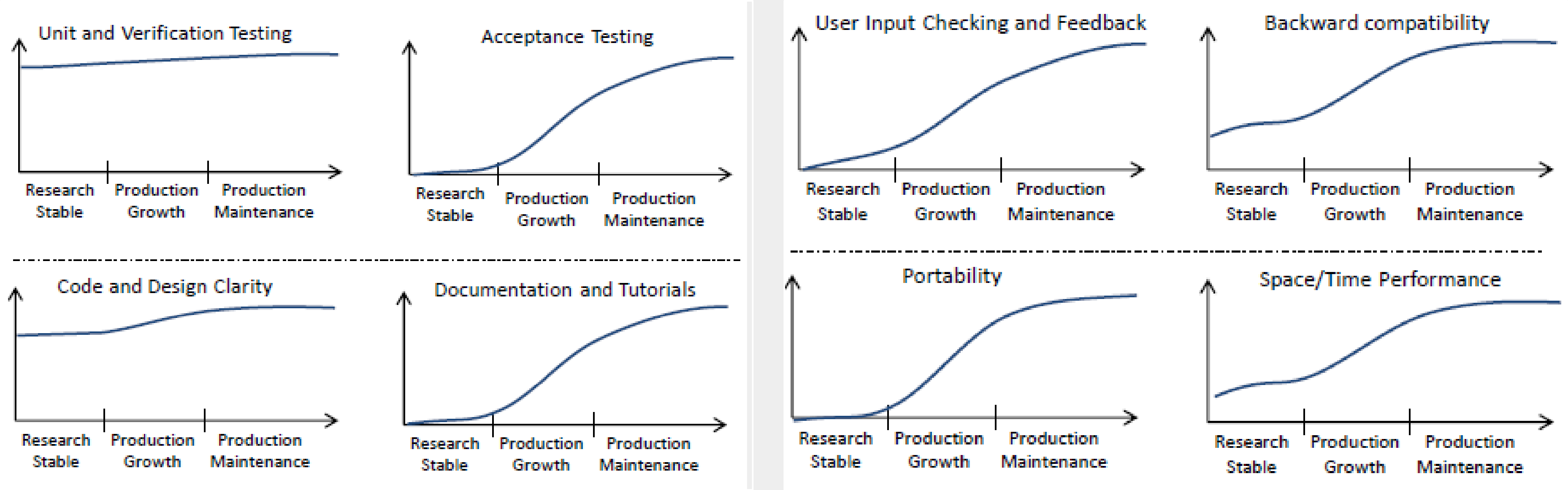


Figure 1. “Example of the more typical variability in key quality metrics in a typical CSE software development process.” From R. Bartlett, et al., “TriBITS Lifecycle Model Version 1.0,” SAND2012-0561, (2012)

Real Software Quality Metrics

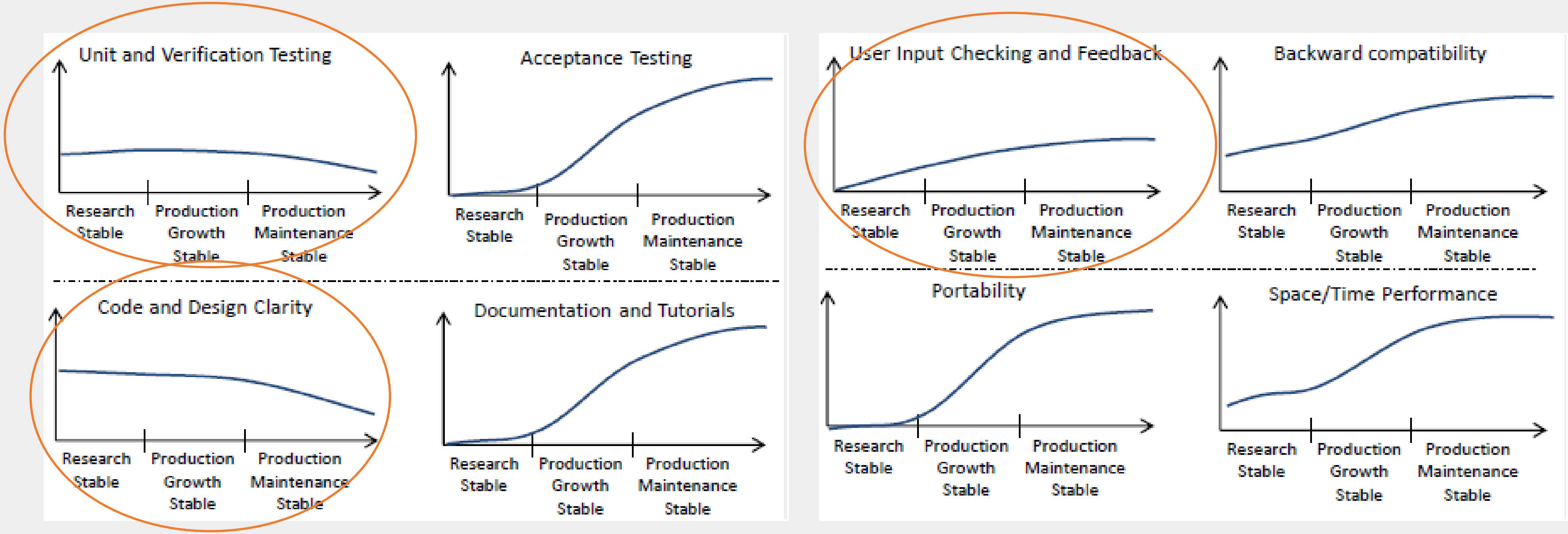
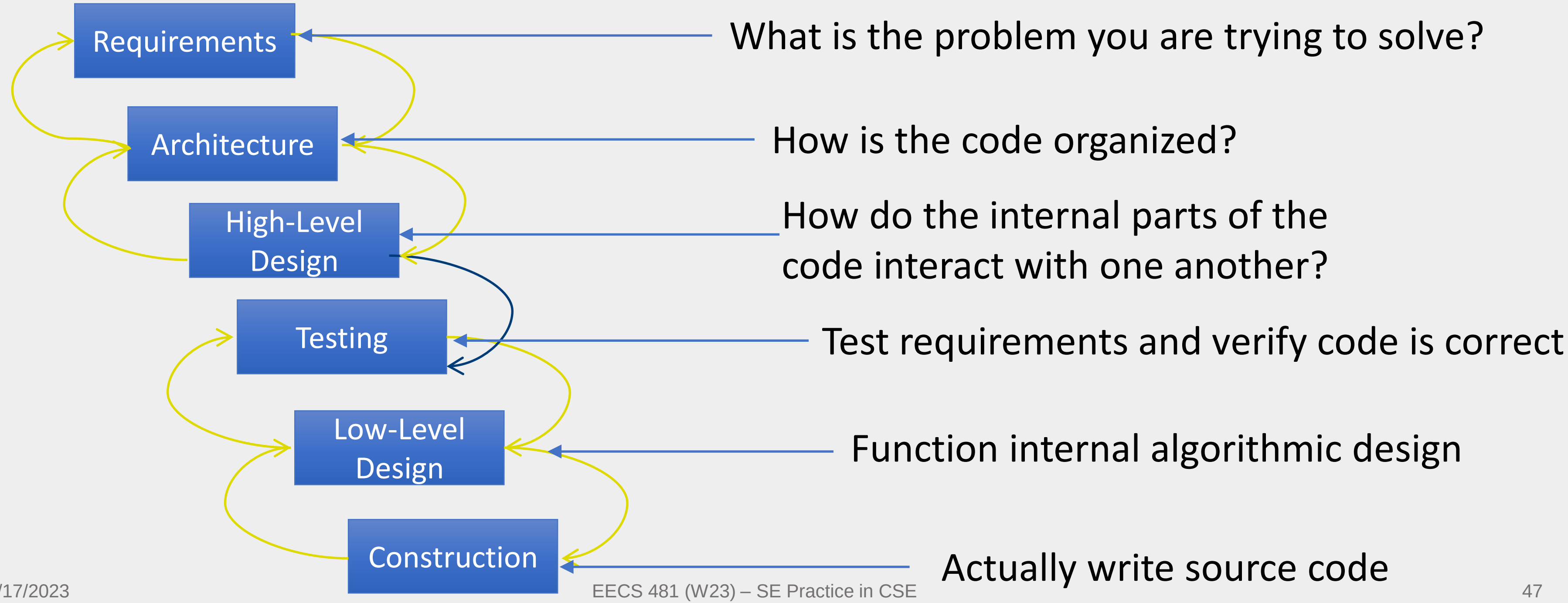


Figure 6. “Example of the more typical variability in key quality metrics in a typical CSE software development process.” From R. Bartlett, et al., “TriBITS Lifecycle Model Version 1.0,” SAND2012-0561, (2012)

The MPACT Dev Process

Iterative development Process



NQA-1 Program

- Sets QA requirements for program
- Each release has
  - Software Management Plan
  - User Manual
  - Theory Manual
  - Verification and Validation Manual
  - Programmer Manual
  - Software Test Plan Requirements and Test Report

Traceability

Req. ID	*Requirement Description* / *Test Name* / *Test Input* / *Additional Info*
1	MPACT shall compute solutions to a 3-D mutliassembly model with control rod movement. MPACT_exe/tests/regression_tests/mini_core/4-mini_apsr.inp
2	MPACT shall compute solutions to a shuffle of an evenpin lattice, mirrored across the x-axis, with 0 quarter rotations. MPACT_exe/tests/regression_tests/solution_verification/xml_input/ shuffle_symmetry/evenpin_mirror-x_rot0.inp
3	MPACT shall compute solutions to a shuffle of an evenpin lattice, mirrored across the x-axis, with 1 quarter rotations. MPACT_exe/tests/regression_tests/solution_verification/xml_input/ shuffle_symmetry/evenpin_mirror-x_rot1.inp
4	MPACT shall compute solutions to a shuffle of an evenpin lattice, mirrored across the x-axis, with 2 quarter rotations. MPACT_exe/tests/regression_tests/solution_verification/xml_input/ shuffle_symmetry/evenpin_mirror-x_rot2.inp
5	MPACT shall compute solutions to a shuffle of an evenpin lattice, mirrored across the x-axis, with 3 quarter rotations. MPACT_exe/tests/regression_tests/solution_verification/xml_input/ shuffle_symmetry/evenpin_mirror-x_rot3.inp

MPACT Software Test Plan, Requirements, and Test Report

CASL

CONTENTS

848	MPACT_exe_testVerify_shuffle_oddpin_rotation-y_rot2	Completed	Passed
849	MPACT_exe_testVerify_shuffle_oddpin_rotation-y_rot3	Completed	Passed
850	MPACT_exe_testVerify_shuffle_oddpin_unfold_mir	Completed	Passed

Defect Reporting: One-Slide Summary

- A software **defect report** includes information and communications related to addressing a software issue.
- Defect reports have many components
- Defect reports are subject to **triage** based on **severity** and **priority** information.
- Defect reports have a lifecycle that is complicated and non-linear with multiple possible **resolutions**.

Defect Reporting and Triage

SOFTWARE PROBLEM REPORT

Problem Report ID: 2020-017-0

Date Received by VSM: March 24

Originator/Originating Organization: EPRI

SOFTWARE PRODUCT: MPACT RELEASE/VERSION #: VERA 4.1RC2

PROBLEM REPORT TYPE

- ☒ Coding Error
- ☐ Documentation
- ☐ Data Library Error
- ☐ Build Issue
- ☐ Other *(specify)*

ATTACHMENTS: ☐ Yes ☐ No

If yes, list attachments:

Error Reporting Category: ☐ Major ☐ Medium ☒ Minor

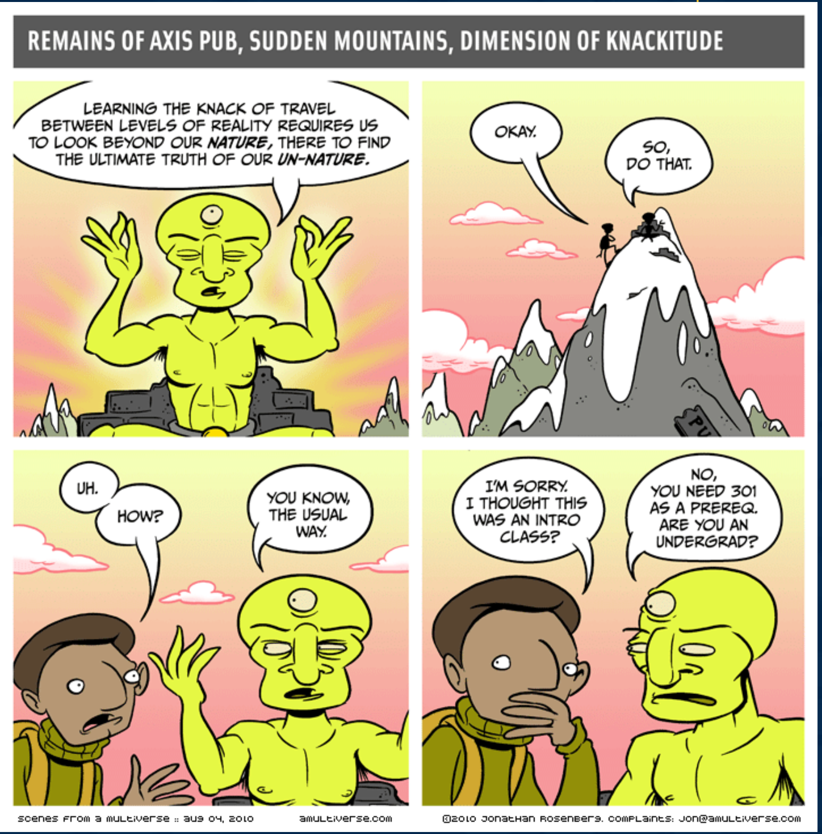
Basis for Error Categorization: ☐ Conveyed by User/Originator ☒ PSM/VSM Evaluation of Error

RELEVANT KANBAN TICKET(S): 6347

Item	Description
Title	*Steady-State Calculations Segfault with Transient MPACT Options*
Data Library	*Unrelated to data libraries*
Description of the error	*If the user is running a steady state calculation and includes either the "prompt" card or the "accel" card in the MPACT block of the input, MPACT crashes with a segfault during input processing.*
How was the error identified?	*A user reported the error*
When does the error occur?	*The error occurs when a user includes either the "prompt" or "accel" card in the MPACT block of their VERA input, but intends to run a steady state calculation. When MPACT begins processing the XML input it segfaults when attempting to deallocate a variable that is already deallocated.*

Potential impact of error	*If users do not realize these cards are present in the input block, then their case will segfault during input processing.*
Frequency/likelihood of error occurring	*This will happen anytime the aforementioned input cards are used in a steady-state calculation.*
How can users determine if error affects their calculations?	*The user can review the MPACT block of their input for presence of the "prompt" or "accel" card.*
What action should users take if error affects them?	*Remove or comment out the "prompt" or "accel" card from the input.*
Is correction to code/data available?	*The code has been modified to ignore these options when present in a steady-state input.*
How to obtain/install correction	*The fix for this will be available in VERA 4.2.*
Additional Comments:	

Test Inputs, Oracles, and Generation



Inputs & Oracles: One-Slide Summary

- Formally, a **test case** consists of an **input** (**data**), an **oracle** (output), and a **comparator**.
- Test inputs determine the behavior of the program. High-coverage inputs can be **generated automatically** through path enumeration, path predicates, and mathematical constraint solving.
- Test oracles correspond to what the program should do. Generating them is an expensive problem; but it can be done automatically (sort of) through invariants and mutation.
- **Test suite minimization** finds the smallest subset of tests that meet a coverage goal.

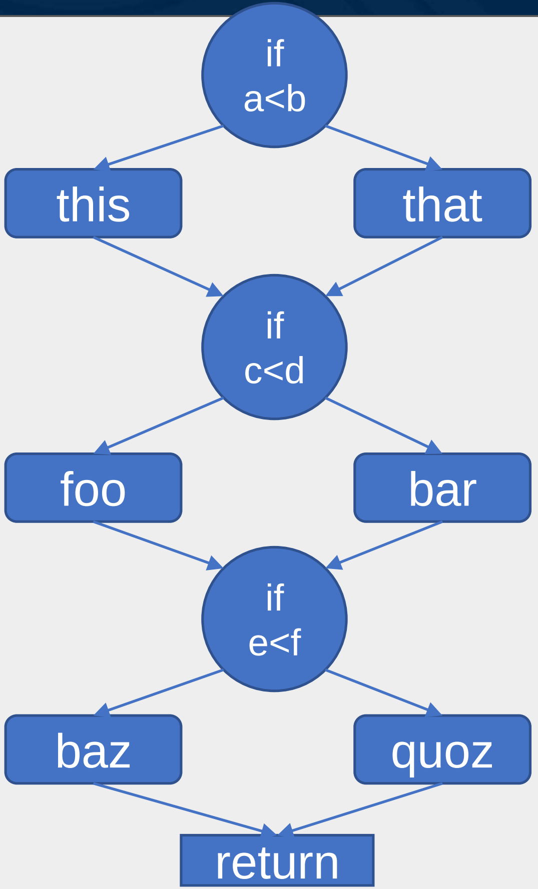
A Taxonomy of Testing (If its not tested, its not a supported feature

- *Unit Testing* – Test individual units of program *in isolation*
  - Should run very fast: < 1 second (a couple seconds is ok)
- *Integral Testing* – Testing program components together
  - Should run fast: < 1 minute (a couple minutes is ok)
- *Regression Testing* – Test whole program for changes in program output
  - Should run fast: < 1 minute (a couple minutes is ok)
- ***Verification Testing* – Test that you are "doing things right"**
  - Can happen at unit or integral or regression level. Comparison analytic solutions or manufactured solutions.
- ***Validation Testing*** – Whole program testing "doing the right thing"; simulating reality, comparison to experiment.
  - May be long running: minutes to hours
- *Memory Testing* – Expensive testing that does detailed memory simulations to detect errors (valgrind)
- *Coverage Testing* – Figure out how much of your source code is actually covered by testing
- *Portability Testing* – test on different platforms and with different compilers

Test input path testing

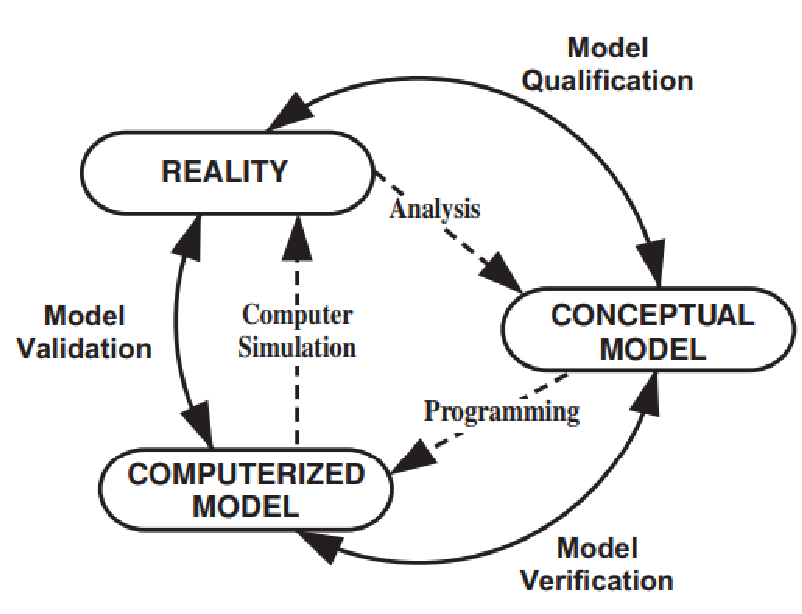
- Change input settings

Solvers	2D P0
	2D Pn
	2D/1D NEM
	2D/1D SP3
	CMFD 1g
	CMFD MG
	Search keff
	Search boron
	Search rod
	Multistate
	Depletion native
	Depletion origen
	Shuffle rotational
	Shuffle mirror
	InternalTH
	CTF
	Eq Xe/Sm
	Subgroup
	ESSM
	Cusping

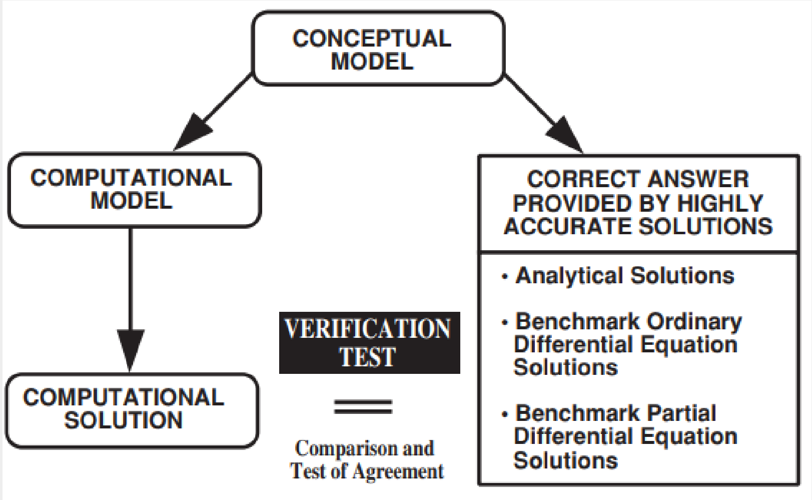


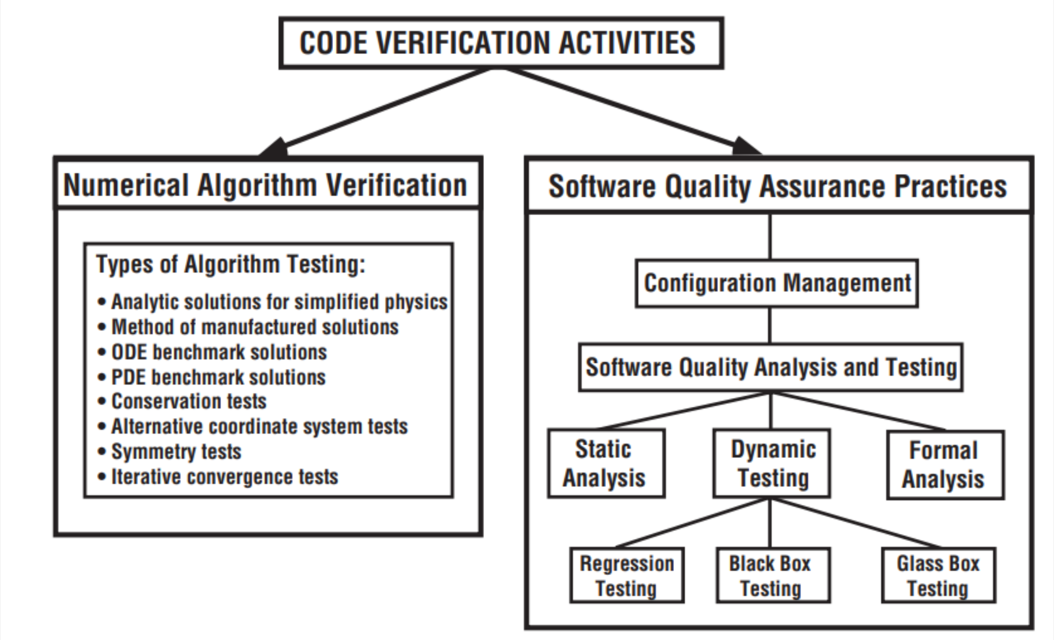
Verification and Validation

- *Model verification*: substantiation that a computerized model represents a conceptual model **within specified limits of accuracy**.
- *Model validation*: substantiation that a computerized model within its domain of applicability possesses a **satisfactory range of accuracy** consistent with the **intended application** of the model.



V&V In the Context of SQA



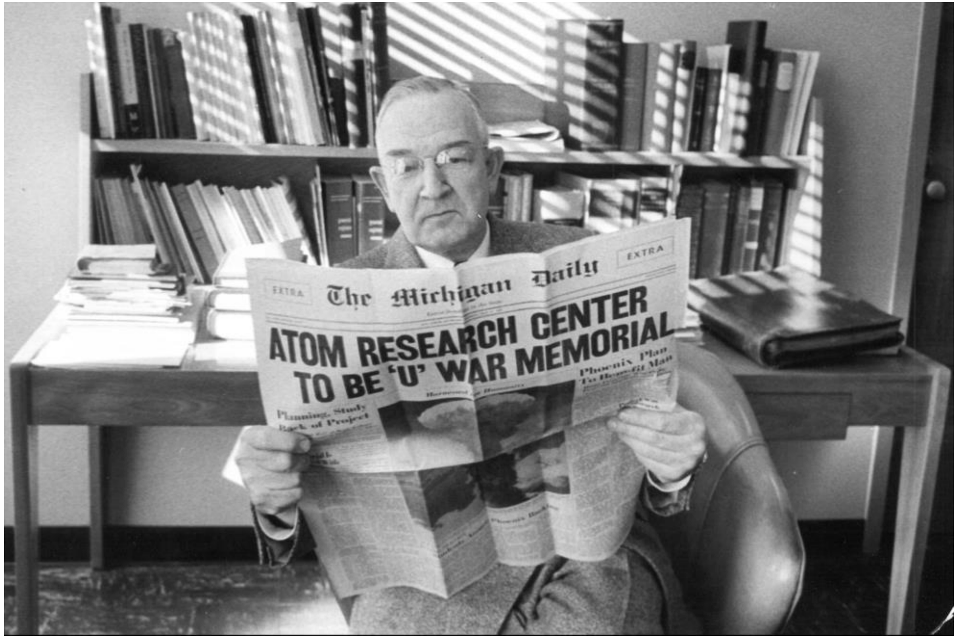


What CSE typically does not do…

- Automatic Test Generation
  - Inputs and oracles
- Mutation analysis
- Path Coverage
- Test Minimization

History of the Ford Nuclear Reactor

Founding
Monday, May 17th, 1948



EXTRA The Michigan Daily EXTRA

Latest Deadline in the State

ATOM RESEARCH CENTER TO BE 'U' WAR MEMORIAL

Planning, Study Back of Project

Year and Half of Work Follows Original Student Legislature Idea

Harnessed for Humanity

Phoenix Plan To Benefit Man

Huge Program Will Probe Peaceful Application of Atom

Memorial Is Greeted With Enthusiasm

Gains Support, Pledges of Aid

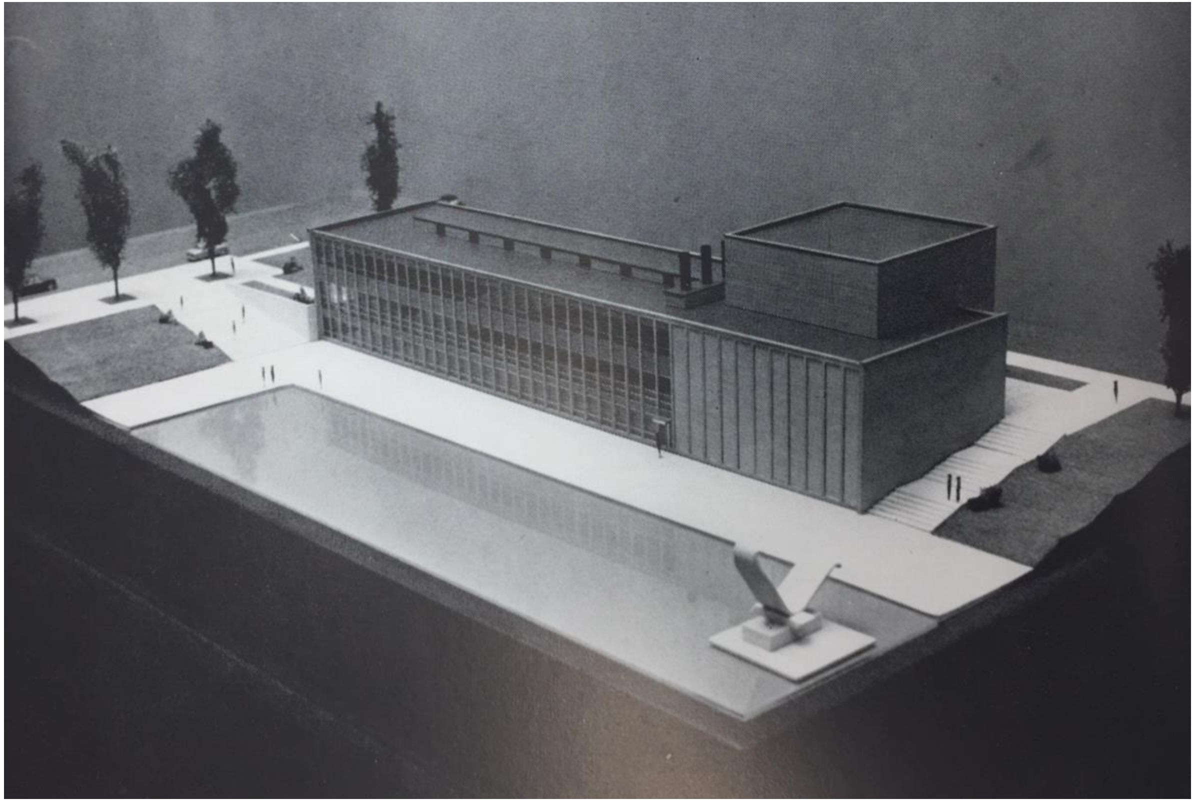
Project Will Aid Research Coordination

War Memorial Is Functional

DYNAMIC REACTION:

Students Assure Phoenix Backing

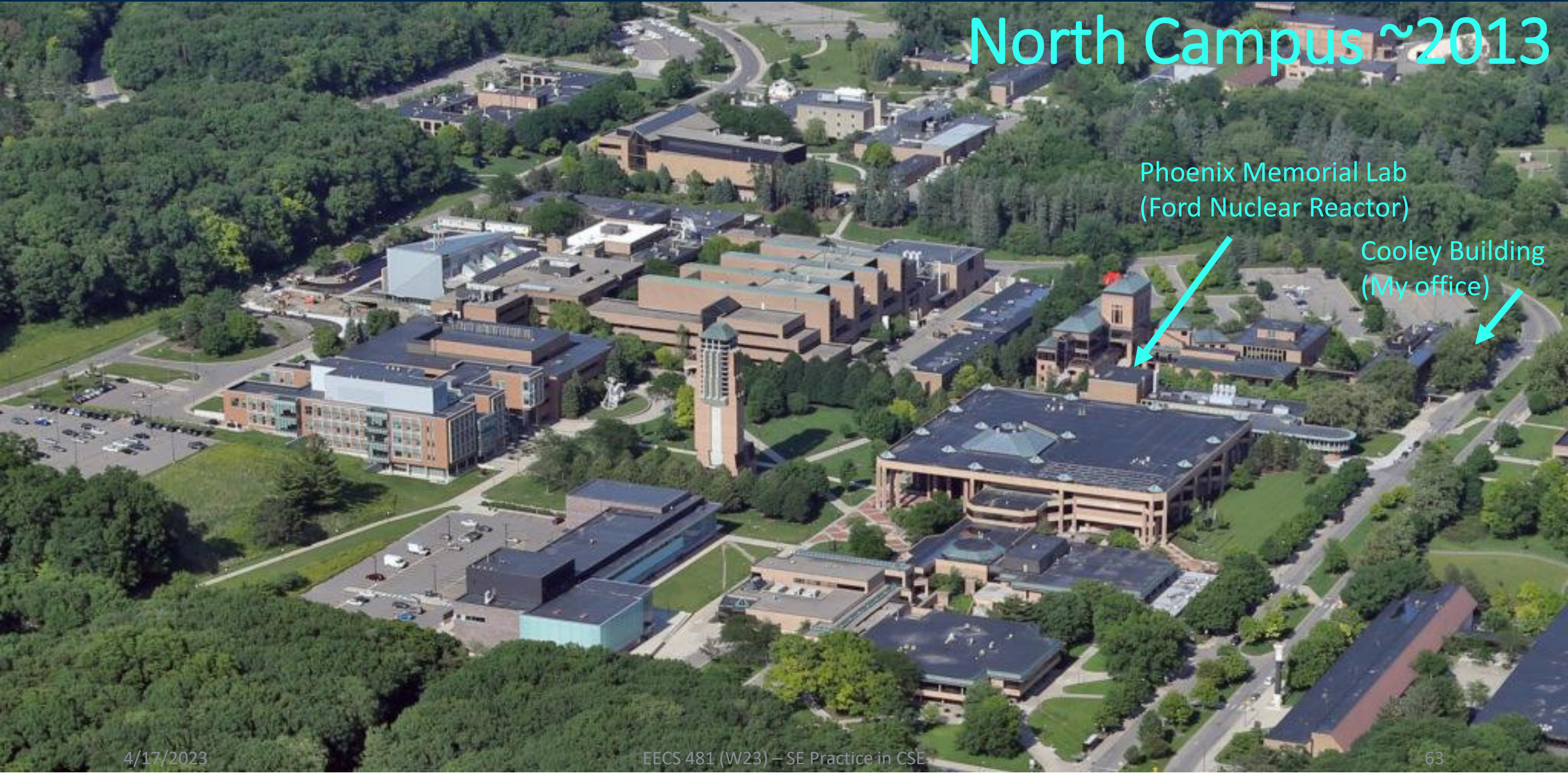
A permanent monument to University dead of World War II, the completed Phoenix Memorial Laboratory will look like this.



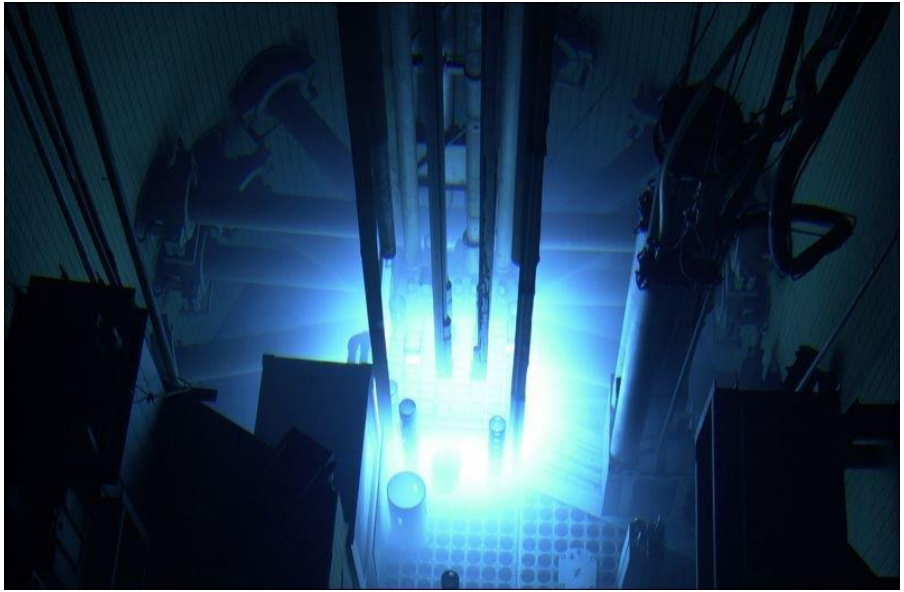
The cubical windowless area at the right represents the ~~student offices~~ nuclear reactor


September 1954
Phoenix Memorial Lab
(Ford Nuclear Reactor)
Cooley Building
(My office)

North Campus ~2013



Ford Nuclear Reactor



- The Ford Nuclear Reactor (FNR) was a research reactor that operated at U of M from September 1957.
- FNR was used to explore peaceful uses of nuclear energy for the well-being and advancement of humanity.
- Unfortunately, FNR was permanently shut down in July 2003 and eventually decommissioned due to the prohibitively high cost of maintenance and operation.

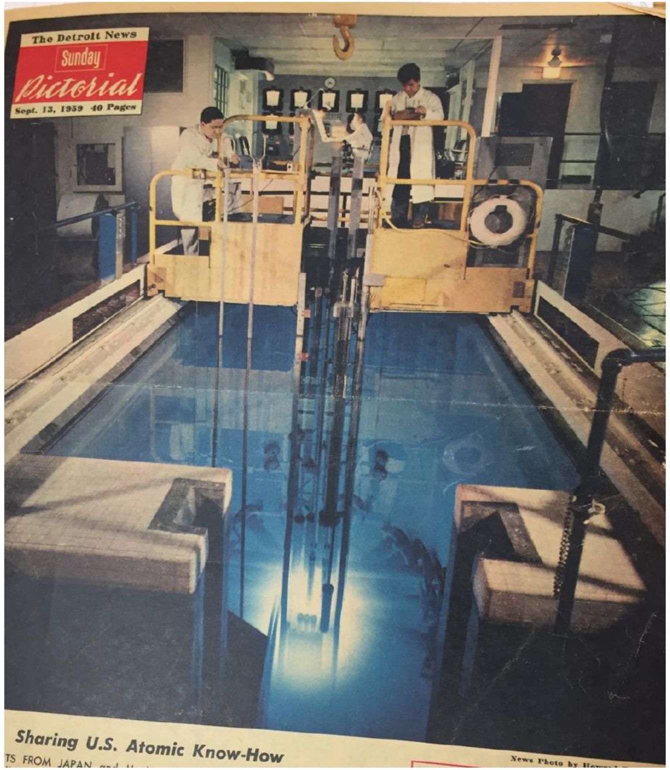


Figure 1. Color photograph of the FNR facility from the Detroit News Sunday Pictorial, Sept. 13, 1959

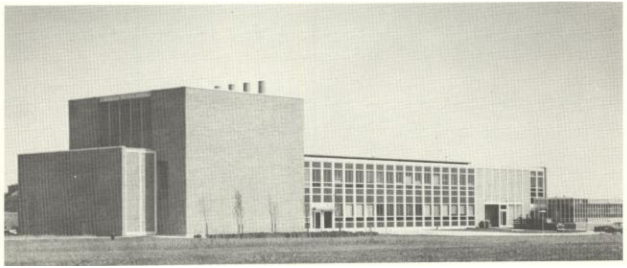


Figure 2. The FNR building and adjacent Phoenix Memorial Lab

Figure 3. Professors in the glow of the reactor plotting their next student assignments

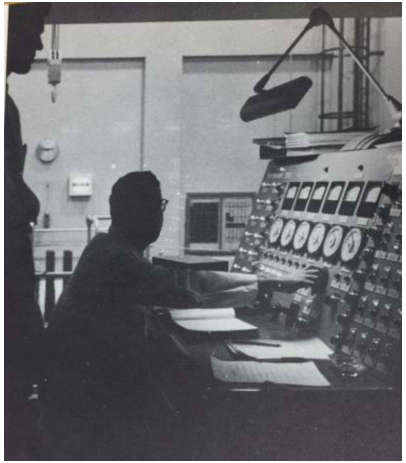


Figure 4. Photograph of a reactor operator at the FNR control panel

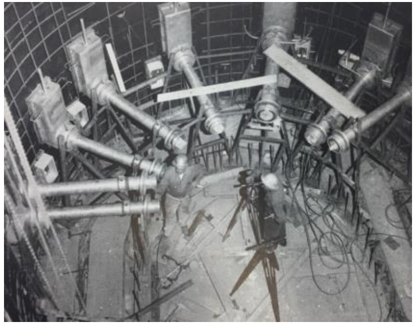
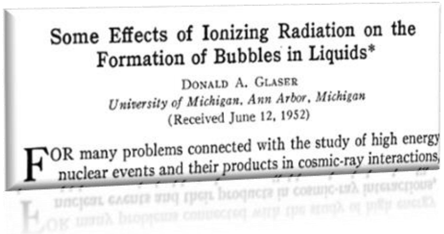


Figure 5. Photograph of FNR reactor pool construction

The first visit to the FNR made by a school group of Burns Park sixth graders taught by Mrs. Betty Melhuish

Mrs. David Weyant demonstrates the versatility of the manipulator by picking up a book of matches, taking one match out, lighting it and then lighting a cigarette for Prof. Ralph A. Sawyer head of the the Phoenix Project



Some Effects of Ionizing Radiation on the Formation of Bubbles in Liquids*

DONALD A. GLASER
University of Michigan, Ann Arbor, Michigan
(Received June 12, 1952)

FOR many problems connected with the study of high energy nuclear events and their products in cosmic-ray interactions,

Fundamental scientific discoveries

Life Sciences

Medicine

Automotive Industry

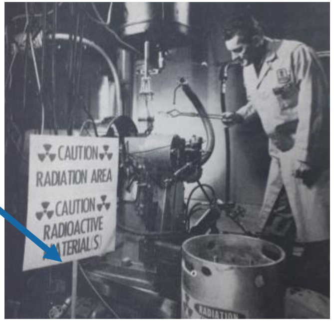
Geology and Archeology

Irradiation to increase shelf-life

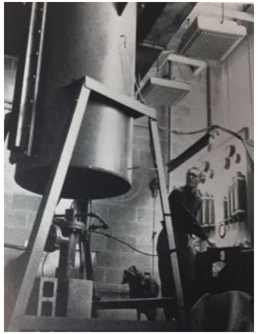
Radioactive tracers in basic study of the ways plants select and absorb nutrients in the soil

Gamma radiation and x-ray on the ability of certain parasites to reproduce

Experiment with Radioisotope tracers

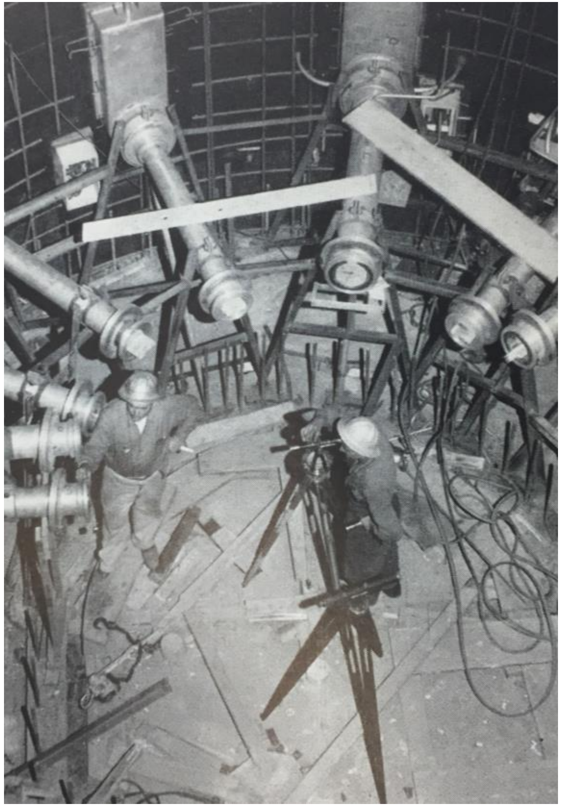


Radioactive isotopes are used in automotive wear studies at GM Technical Center



The effects of alpha radiation on the combustion of gaseous fuels

One of the largest representative collections in the country of radioactive ores has been gathered in the Department of Geology



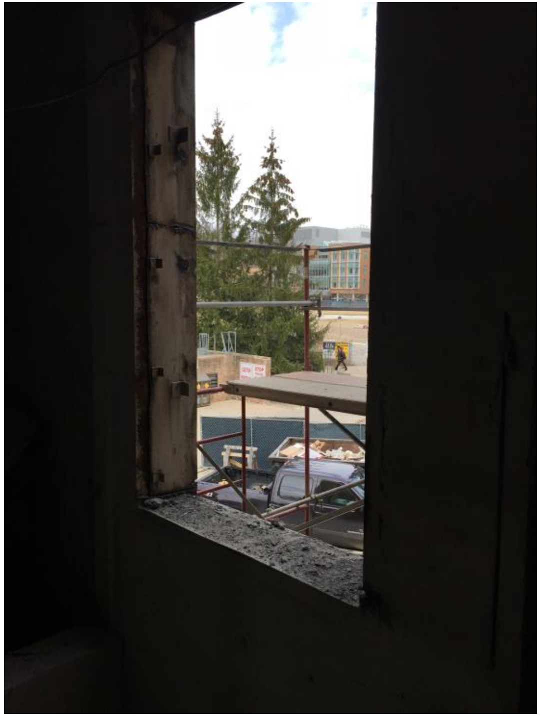
1950's

2016

Filling in the FNR Pool with Concrete

Cutting Windows in the Reactor Building

Let there be light!
And so the students shall have windows.



2016

Timeline of VFNR Development

9/2019

CAD Model Development

Information Gathering

- Interviews
- Library visits
- Extensive Web searches
- Digitization of hardcopies

1/2020

Experiment Design

Review of historical syllabus

- Old lab reports
- Learning Pedagogy in VR
- Revise experimental procedures for VR
- Story Boarding

Everyone went on vacation for a year*

5/2021

Integrated Simulation

Simulation Development

- FNR Dynamics model development
- Testing of FMU Export
- Testing of Unreal Engine FMU Plug-in
- Interactive capabilities

7/2022

Immersiveness

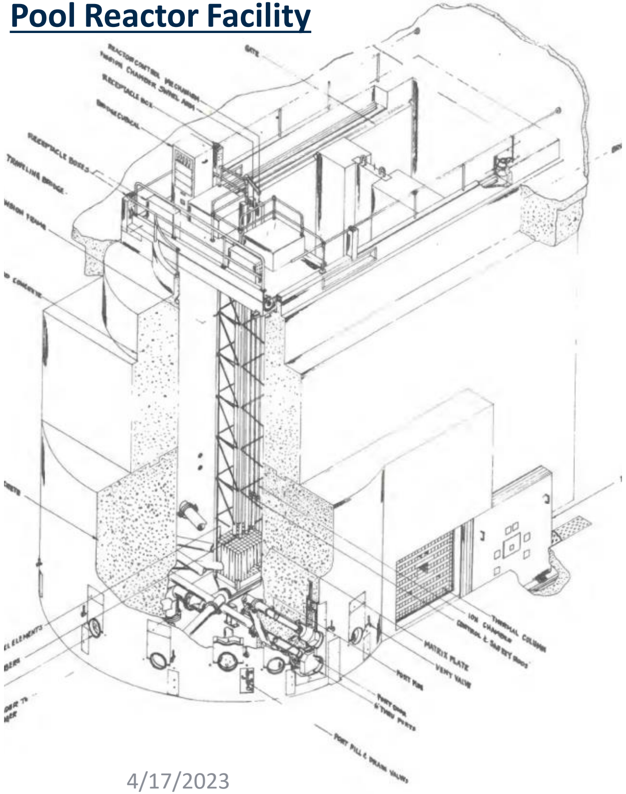
Art & Design

- Objects
- Lighting
- Textures
- Audio

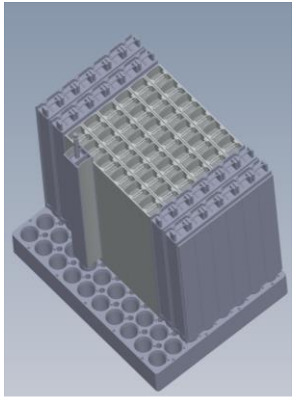
10/2023

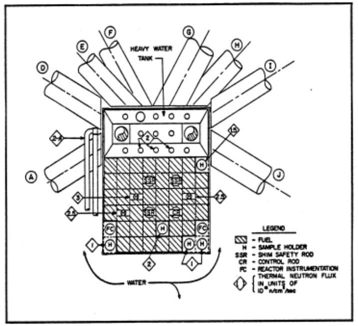
Project Phase 1: CAD Model Development

Pool Reactor Facility



Nuclear Core





Nuclear Fuel Element